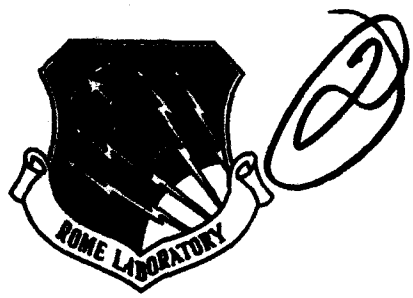
RL-TR-91-353
Final Technical Report
December 1991






# SMART BIT/TSMD INTEGRATION

Grumman Aerospace Corp.

Charles Cooper, Kenneth A. Haller, Victor G. Zourides,
Paul Skeberdis, White Gibson



*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*



Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-353 has been reviewed and is approved for publication.

APPROVED:

Dale W. Richards

DALE W. RICHARDS
Project Engineer

FOR THE COMMANDER:

John J. Bart

JOHN J. BART
Deputy Director
Electromagnetics and Reliability Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(ERSD), Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



| 1. AGENCY USE ONLY (Leave Blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | December 1991 | Final Sep 89 - Jan 91 |

**4. TITLE AND SUBTITLE**
SMART BIT/TSMD INTEGRATION

**5. FUNDING NUMBERS**
C - F30602-89-C-0155
PE - 62702F
PR - 2338
TA - 02
WU - 4K

**6. AUTHOR(S)**
Charles Cooper, Kenneth A. Haller, Victor G. Zourides,
Paul Skeberdis, White Gibson

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Grumman Aerospace Corp.
ILS Advanced Development, C10-106
Bethpage NY 11714

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Rome Laboratory (ERSD)
Griffiss AFB NY 13441-5700

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
RL-TR-91-353

**11. SUPPLEMENTARY NOTES**
RL Project Engineer: Dale W. Richards/RL(ERSD)/(315)330-3476
Subcontractor: Westinghouse Electric Corp.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (Maximum 200 words)**

The problem of false alarms and intermittent failures of electonics continually manifest themselves in high levels of Cannot Duplicate (CND) and Retest OK (RETOK) reports. These conditions lead to an excessive drain on resources including spares, manpower and test equipment and ultimately produce a negative impact on mission readiness. A major contributor of false alarms/intermittents is that of external environmental factors. Smart Built-in Test (Smart BIT) uses environmental stress data as inputs to its artificial intelligence based reasoning process for detecting false alarms, and Time Stress Measurement Device (TSMD) technology provides a capability to measure, collect and store stress data for failure correlation, although with minimal attention to date to real-time applications

Until now, these technologies were independent research and development activities. This report discusses the integration of these technologies. A self contained laboratory testbed was developed to support this research. Hardware and software computer design and development necessary to investigate and complete the integration are detailed. Methods for simulating the environment based on real world data as well as actual sensor hardware implementations were developed. Guidelines for installing actual electronics in the testbed are provided along with conclusions and recommendations for fielding an integrated capability.

**14. SUBJECT TERMS**
Built-in Test, Intermittent, Time-Stress Measurement Device, Artificial Intelligence, Cannot Duplicate, False Alarm, Retest Okay

**15. NUMBER OF PAGES**
202

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

# EXECUTIVE SUMMARY

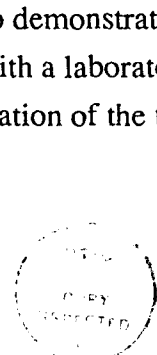
## INTRODUCTION

The problems of false alarms and intermittent failures continuously manifest themselves in varying degrees in avionics equipment and their suppression/elimination is a continuing effort. These conditions lead to two symptoms with serious detrimental effects: Cannot Duplicate (CND) and ReTest OK (RTOK). The impact of these problems leads to an excessive drain on resources, including spares, personnel, publications, training, test equipment, and ultimately on mission readiness.

The reasons for false alarms are varied and include such factors as environment (temperature, humidity, vibration, etc), electromagnetic interference, equipment degradation due to age, tolerance design factors, maintenance-introduced factors (e.g., connectors, wire handling), combinations of the above, and other anomolous causes.

Rome Laboratory has addressed this problem via two thrusts: Smart Built-In-Test (BIT) and Time Stress Measurement Device (TSMD). Smart BIT adds Artificial Intelligence (AI) techniques to BIT to make the tests far more robust in determining the cause of the false alarm, instead of just a simple determination based on a GO/NO-GO trigger mechanism. Smart BIT uses real-time environmental stress data for its reasoning process. TSMD efforts have concentrated on the measurement, collection, storage, and failure correlation analysis of stress data with minimal attention to the real-time use of that capability. External environmental factors cause overstress of avionics equipment and are therefore believed to be a leading cause of false alarms.

Rome Laboratory has determined that the integration of both of these technologies, Smart BIT and TSMD (or Smart BIT/TSMD), will enhance the detection and determination of false alarms. The objective of this effort is to develop a testbed to demonstrate and assess such an integration. Smart BIT/TSMD provides Rome Laboratory with a laboratory testbed to evaluate and assess the individual characteristics as well as the integration of the two technologies.



## BACKGROUND

Smart BIT/TSMD involves the integration of two technologies, Smart BIT and TSMD, developed by Grumman and Westinghouse, respectively. Smart BIT is based on two earlier programs conducted by Grumman under contract to Rome Air Development Center (RADC). The first of these, Smart BIT, Contract No. F30602-84-C-0051, detailed in Reference 1 and summarized in Reference 2, successfully proved that AI could be used to discriminate false alarms from hard faults. The follow-on contract to Smart BIT, termed Smart BIT-2 (RADC Contract No. F30602-86-C-0272), detailed in Reference 3 and summarized in Reference 4), was also awarded to Grumman. Its purpose was to prove that the AI techniques developed in the first Smart BIT contract, with enhancements, could be applied to a modern system – one utilizing microprocessors, VLSI, BUS, and digital and analog circuitry. In addition Grumman, under contract to RADC, studied the feasibility of incorporating Smart BIT techniques into portions of the Joint Surveillance Target Attack Radar System (Joint Stars). This study, Smart BIT for Joint Stars (Contract No. F30602-87-C-0174) is detailed in Reference 5. Reference 6 summarizes the above Smart BIT programs.

Westinghouse, under contract to RADC for the Design Definition Phase for MicroTime Stress Measurement Device (TSMD) Development (F30602-87-C-0152), developed and demonstrated a TSMD capability in a hybrid microelectronic package.

## SMART BIT/TSMD SYSTEM OVERVIEW

The Smart BIT/TSMD Integration System is a self-contained laboratory test bed. The test bed utilizes commercially available microcomputers to host both the Smart BIT and TSMD functions and provide interesting and representative data to stimulate those techniques. It consists of four major hardware subsystems: the UUT computer, the LRU computer, the TSMD computer, and the Smart BIT computer. The Smart BIT/TSMD Integration overall system block diagram is shown in Figure 1.

The LRU computer assembly has its own power supply, a contractor-developed sensor board, and the capability to emulate BIT data and traffic. It is used to simulate a Line Replaceable Unit (LRU) and is modified by adding TSMD sensors (accelerometer and temperature sensors). Faults are simulated simultaneously with a corresponding change in temperature and/or acceleration, to replicate false alarms and intermittent BIT failure reports. The UUT computer sends

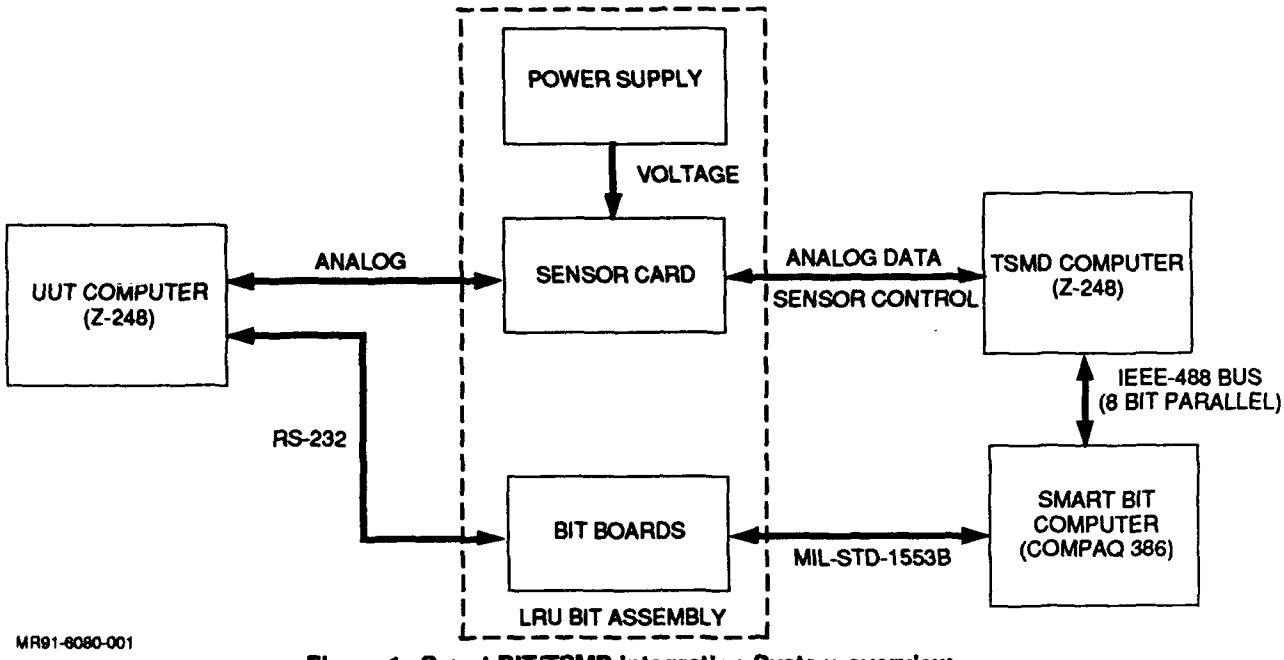



**Figure 1. Smart BIT/TSMD Integration System overview**

commands to the LRU computer's BIT to indicate OK or fault conditions and simulate the sensor signals. The sensor card conditions sensor data for the TSMD computer for further analysis, data compression, and datalogging. The Smart BIT computer periodically interrogates both the LRU and the TSMD computers. It then analyzes both data sets, and BIT and sensor information, using AI techniques to determine the UUTs true status (OK, intermittents, hard faults). Emphasis is placed on integrating the Smart BIT and the TSMD technologies while maintaining the independence of the individual techniques. To perform the Smart BIT and/or TSMD assessment, the testbed operator selects the scenario from the UUT computer that simulates stress and/or BIT reports and examines the outputs of the TSMD and Smart BIT computers as the scenario runs.

Assessment of the Smart BIT techniques and the integration of TSMD sensor data are shown both in a graphical and textual manner. This includes strip-chart windows of the Smart BIT techniques as well as an indicator window that allows direct comparison between Smart BIT and normal operation BIT results. The output of the simulated LRU, combined with the TSMD sensor data, is analyzed by the Smart BIT subsystem using several Smart BIT techniques to classify LRU fault behavior. An additional window is provided that indicates the Smart BIT conclusion about the UUT and the confidence of that assessment.

## INTEGRATION FACTORS

In order to integrate Smart BIT and TSMD, various items are required. Obviously Smart BIT and TSMD are among them. In addition, viable data, on which both Smart BIT and TSMD are to operate, are essential. Otherwise, the entire basis of their integration would be meaningless. These data are used to build real-world scenarios for simulating faults of an LRU's BIT circuitry simultaneously with temperature and/or acceleration. As a result, false alarms/intermittent BIT failure reports can be studied along with environmental changes. The final ingredient, then, is a suitable LRU to provide interesting and representative data with which to stimulate the above integrated system.

### Aircraft Environmental Data

Temperature and vibration data from the F/FB-111 Tail Pod Assembly were used as the source for developing environmental scenarios. Vibration and temperature data were derived as discussed in detail in Subsection 2.1.

For vibration, the following flight mode scenarios were simulated: speed brakes out, normal level flight, level turn constant g, level turn max g, weapons drop, and side slip. BIT failures were correlated with the peak values of vibration.

For temperature, BIT faults were correlated with scenarios where temperature either exceeds a high limit, but not critically, and cycles at a rate considered reasonable.

### AI Techniques

The AI techniques used were based on the previous Smart BIT and Smart BIT-2 programs. These included Adaptive BIT, Information Enhanced BIT, Improved Decision BIT, and Temporal Monitoring software techniques. Adaptive BIT and Temporal Monitoring techniques are refinements of Improved Decision BIT and Information Enhanced BIT which are general methods for making better BIT evaluation decisions.

Two different approaches of Adaptive BIT were implemented:

- Neural Network Paradigm. The network trains on known intermittencies over a time period, adjusting its interconnecting weights so that it properly classifies new unknown behavior. In essence, it self-adapts to changing levels of induced faults
- K-Nearest Neighbor. The unknown behavior is compared to its nearest neighbors and a majority vote determines its behavior.

Rather than reporting out an immediate HARD fault, the temporal monitor makes a transition to an intermittent state and continues to monitor in time. Transition probabilities are estimated in real time. These estimates are used as the criteria for transitioning from an intermittent state to the OK or HARD fault state.

### TSMD Techniques

The TSMD computer provides the capability of performing, assessing and evaluating the TSMD processing algorithms. These algorithms include life-stress monitoring, over-stress event logging, and fault event logging. The following functions are performed:

- Interface with the sensor board in the LRU BIT computer
- Perform TSMD sensor data collection, processing, compaction, and storage functions
- Display TSMD data to the operator
- Interface with the Smart BIT computer.

The TSMD software manages the two communications interfaces to provide the required real-time link between the TSMD sensor card and the Smart BIT computer. Display software provides the operator with the capability to visually assess the performance of the TSMD data collection, processing algorithms, and sensor data storage.

### Equipment Selection

The Standard Central Air Data Computer (SCADC) was selected as the vehicle for providing interesting data to stimulate the Smart BIT and TSMD techniques, principally because it is representative of a cross-section of technologies found in modern avionic systems. It contains a digital microprocessor, semiconductor memory, random logic, and analog circuitry. It was also chosen because of the amount of Re-Test (RTOK) maintenance actions that were recorded on various F-111 aircraft. In addition, the SCADC BIT circuitry was investigated in the Smart BIT-2 contract for developing Smart BIT techniques.

## TEST BED DESCRIPTION

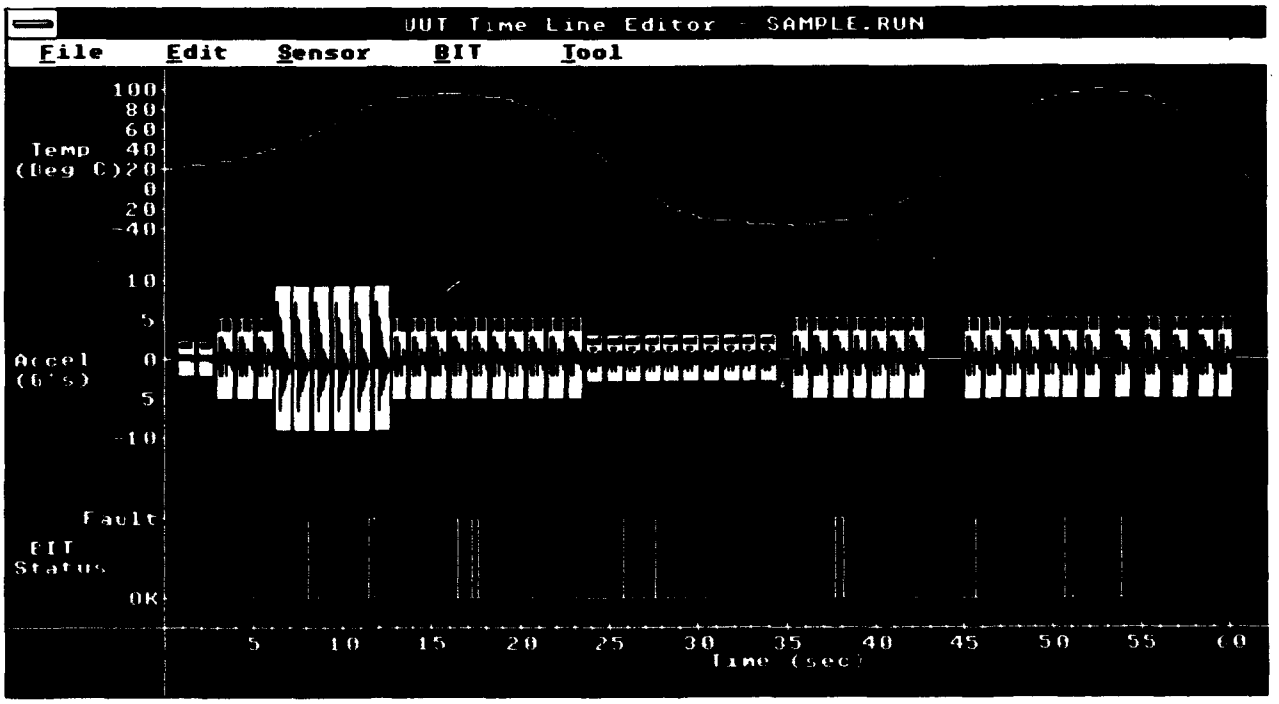
### UUT Computer

The UUT computer contains programs to create typical BIT reports and sensor data profiles and send them to the sensor card and Smart BIT computer. A timeline editor allows the operator to populate the data file with UUT BIT as well as sensor data simulation. The UUT computer controls the operation of the UUT by sending commands to indicate fault conditions to the LRU

BIT computer and, concurrently, to the sensor card during scenario operation. The UUT computer sends, via the digital interface, a command message to the LRU BIT computer to indicate a selected BIT fault. This command message contains the scenario selection and the fault mode (fault modes include FAULT and OK). The LRU BIT computer responds to a fault command by setting up its MIL-STD-1553B data tables with the appropriate faulted data and BIT status (information read periodically by the Smart BIT computer for analysis). The LRU BIT computer responds to an OK command from the UUT computer by returning its MIL-STD-1553B data tables and BIT status to normal (no fault) data.

When the scenario requires sensory-caused faults, the UUT computer sets (concurrently with the LRU BIT command) the appropriate analog output signal to a level that will be beyond the acceptable LRU specification range for that sensor type. This sensor data will be read by the TSMD computer for processing.

The UUT computer provides the scenario time line editor (Figure 2) to generate data files for controlling the TSMD sensor card as well as the simulation of normal, intermittent, and hard fault BIT behaviors. The UUT computer software provides an easy user interface to generate the scenario, BIT, and TSMD data files.




R91-6080-002

**Figure 2. Time line editor.**

LRU BIT Computer

The LRU computer consists of a complete VME Bus computer, enabling the MIL-STD-1553 bus emulation of virtually any LRU of modern Air Force avionics equipment. For the Smart BIT/TSMD integration program, a Standard Central Air Data Computer (SCADC) is emulated. When a SCADC is requested to report its output message data block by the General Navigation Computer (GNC) in an actual system, it sends back data similar to the parametric information shown in Table 1 over the MIL-STD-1553B bus. In the case of the Smart BIT/TSMD system, the role of the GNC is simulated by the Smart BIT computer and the SCADC by the LRU computer. The LRU BIT computer therefore responds to periodic requests for data by the Smart BIT computer in much the same way as would an actual SCADC. If the LRU BIT computer is commanded by the UUT computer to simulate a selected fault scenario, the parametric information that the LRU computer reports might appear as the data in Table 2.

**TABLE 1. SCADC All Outputs Fixed Scenario (No Fault Data).**

| WORD NO. | DATA (HEX) | DESCRIPTION | DATA (DECIMAL) |
|---|---|---|---|
| 1 | FF00 | BIT WORD | 65280.0 |
| 2 | 1310 | PRESSURE ALTITUDE | 12200.0 FT |
| 3 | 1310 | BAROSET ALTITUDE | 12200.0 FT |
| 4 | 15E0 | TRUE AIRSPEED | 350.0 KNOTS |
| MR91-6080-003 | | | |

**TABLE 2. SCADC All Outputs Fixed Scenario (Fault Condition Data).**

| WORD NO. | DATA (HEX) | DESCRIPTION | DATA (DECIMAL) |
|---|---|---|---|
| 1 | 7B00 * | BIT WORD | 31488.0 |
| 2 | 0000 * | PRESSURE ALTITUDE | 0.0 FT |
| 3 | 1310 | BAROSET ALTITUDE | 12200.0 FT |
| 4 | 15E0 | TRUE AIRSPEED | 350.0 KNOTS |
| *NOTE: PARAMETRIC DATA THAT IS MARKED INDICATES FAULTED DATA | | | |
| MR91-6080-004 | | | |

In addition, the LRU BIT computer contains the sensor card that conditions the temperature and accelerometer sensor signals for the TSMD computer. The TSMD computer formats, filters, timetags, compresses, and datalogs the conditioned sensor data from the sensor card.

In summary, the LRU BIT computer continuously simulates BIT reports. It responds to a UUT command to indicate a selected fault condition, according to the selected scenario. This condition affects the operational data as well as the BIT data. This BIT status is read by the Smart BIT computer.

TSMD Computer

The TSMD computer collects conditioned data from the temperature and accelerometer sensors. These signals may be selectively superimposed on simulated sensor signals from the UUT computer. When data are requested by the Smart BIT computer, the TSMD computer provides the conditioned sensor signal data. Over-stress algorithms record events precipitated by a thresholded sensor reading. The data saved under these algorithms may be used for trending, exceeding forecasting, and specific condition monitoring.

Display software provides the operator with the capability to visually assess the performance of the TSMD data collection, processing algorithms, and sensor data storage.

Smart BIT Computer

The Smart BIT computer runs common LISP under the Unix System V operating system. It contains Adaptive BIT, Information Enhanced BIT, Improved Decision BIT, and Temporal Monitoring software techniques – all written in Common LISP. Improved Decision BIT and Information Enhanced BIT are general methods for making better BIT evaluation decisions. Adaptive BIT and Temporal Monitoring techniques are refinements of Improved Decision BIT and Information Enhanced BIT. Therefore, discussions throughout this document that refer to Adaptive BIT and Temporal Monitoring techniques are also referring to Improved Decision BIT and Information Enhanced BIT but in a more focused and specific manner. The Smart BIT computer contains the device drivers, written in "C", and test software for the Commercial-Off-the-Shelf (COTS) interface cards. These interface cards are used to communicate with the LRU BIT computer, via MIL-STD-1553B protocol, to obtain the BIT status and the IEEE-488 parallel bus to communicate with the TSMD computer, for the TSMD data. The Smart BIT computer then uses the above Smart BIT techniques to evaluate the UUT status. The Smart BIT computer also contains software to allow file storage and retrieval of the scenario data. The Smart BIT computer graphically displays the Smart BIT techniques, the BIT input from the LRU BIT computer, and the results of the Smart BIT reasoning processes, as well as an indication of UUT health and the confidence associated with that assessment. These windows are color-keyed to tie together multiple windows for the same Smart BIT techniques. The display of the neural network average error is also provided to illustrate network learning, (when the neural network is chosen for the Adaptive BIT technique). In addition, it displays the LRU BIT report, a typical Operational Flight Program (OFP) response to the LRU BIT report and the Smart BIT status in a lamp indicator form similar to a pilot's console lamp indicator, to allow easier comparison of the

Smart BIT results with the present BIT. A readiness status test may also be run from the Smart BIT computer to check the Smart BIT/TSMD Integration System as a whole.

## INTEGRATION DESIGN CRITERIA

In order to achieve integration of the Smart BIT and TSMD functions, certain issues had to be resolved. These included such issues as how fast data is to be transferred between Smart BIT and TSMD, what types of information must be passed to and from each subsystem, and how information that is passed between each subsystem is integrated into its internal processes. The Smart BIT/TSMD test bed answered these issues based on actual avionic LRU rates, bus protocols based on industry standards, and the real-time exchange of information between the Smart BIT, LRU BIT, and TSMD computers.

## INTEGRATION SCENARIO

Three demonstration scenarios have been provided with the Smart BIT/TSMD Integration System. All three scenarios were obtained from the SCADC FMECA (Failure Mode, Effect, and Criticality Analysis) reports. The scenarios chosen illustrate the Smart BIT/TSMD integration techniques by providing three distinctly different temperature, vibration, and BIT time lines, as well as LRU scenario data sets based on SCADC LRU FMECA failure modes. All three scenarios used the UUT computer to simulate the temperature and accelerometer sensor output, in lieu of the actual sensors, for the purpose of repeatability.

## CONCLUSIONS

- The objective has been met in providing a test bed that may be used to integrate Smart BIT and TSMD techniques and ultimately transitioning these techniques from the laboratory to the field
- The test bed allows the further development of TSMD and Smart BIT techniques, including:
  - Different TSMD algorithms for preprocessing and data compression
  - Building on present Smart BIT techniques for more effective analysis
  - Flexibility to continually integrate the various features of the above techniques
- The test bed allows for the study and integration of Smart BIT and TSMD technologies while retaining their independent capability
- Environmental stresses may be simulated or induced by actual sensors, or a combination of both used for experimentation

- The test bed is not limited to a single avionic LRU but is capable of simulating any 1553B bus-related LRU
- Software modifications can be made without external support. All development tools reside on the test bed
- The test bed simulates a real-world environment, derived from actual data captured from in-flight sensing
- The test bed provides for the use of new field data as they become available.

## RECOMMENDATIONS

- The studies conducted to date have proven that AI techniques can be added to BIT to identify false alarms. However, they have been conducted under lab conditions with sophisticated computers and high-level software tools. A final application study should be conducted to reduce this sophistication and complexity to onboard avionics
- Test Bed Augmentation
  - Substitute an actual LRU avionics unit for the LRU/BIT computer
  - Replace the present Smart BIT computer with a faster processing computer, such as a SUN workstation, or an enhanced Macintosh with a MAC Ivory or micro-Explorer card to obtain faster processing speeds
  - Provide display modifications to enhance graphic/data integration
- Conduct a study to integrate the benefits of neural net adaptive learning (to associate BIT fault patterns with environmental stress data) and the ability to track BIT behavior over time (temporal monitoring) to find the best combination of speed and classification performance
- Conduct a study to use TSMD historical data with neural nets
- Investigate the potential for using hardware NN ICs for implementing selected paradigms
- Develop a Smart BIT card for advanced avionics applications, eventually leading to a Smart BIT chip
- Develop an avionic Smart BIT/TSMD specification for incorporation of these technologies into advanced avionics.

## CONTENTS

## CONTENTS (Contd)

## CONTENTS (Contd)

## ILLUSTRATIONS

## ILLUSTRATIONS (Cont'd)



## TABLES

# 1 – INTRODUCTION

## 1.1 BACKGROUND

Smart BIT/TSMD involves the integration of two technologies, Smart Built-In-Test (BIT) and Time Stress Measurement Device (TSMD), developed by Grumman and Westinghouse, respectively. Smart BIT is based on two earlier programs conducted by Grumman, under contract to Rome Air Development Center (RADC). The first of these, Smart BIT, Contract No. F30602-84-C-0051 detailed in Reference 1 and summarized in Reference 2, successfully proved that Artificial Intelligence (AI) could be used to discriminate false alarms from hard faults. The follow-on contract to Smart BIT, termed Smart BIT-2 (RADC Contract No. F30602-89-C-0272), detailed in Reference 3 and summarized in Reference 4, was also awarded to Grumman. Its purpose was to prove that the AI techniques developed in Smart BIT, with enhancements, could be applied to a modern system – one utilizing microprocessors, VLSI, BUS, and digital and analog circuitry. In addition Grumman, under contract to RADC, studied the feasibility of incorporating Smart BIT techniques into portions of the Joint Surveillance Target Attack Radar System (Joint Stars). This study, Smart BIT for Joint Stars (Contract No. F30602-87-C-0174) is detailed in Reference 5. A summary paper describing the above Smart BIT programs is presented in Reference 6.

Rome Laboratory has addressed this problem via two thrusts: BIT and TSMD. Smart BIT adds AI techniques to BIT to make the tests far more robust in determining the cause of the false alarm, instead of just a simple determination based on a GO/NO-GO trigger mechanism. Smart BIT uses real-time environmental stress data for its reasoning process. TSMD efforts have concentrated on the measurement, collection, storage, and failure correlation analysis of stress data with minimal attention to the real-time use of that capability. External environmental factors cause overstress of avionics equipment and are therefore believed to be a leading cause of false alarms.

Westinghouse, under contract to RADC for the Design Definition Phase for MicroTime Stress Measurement Device (Micro-TSMD) Development (F30602-87-C-0152), developed a hybrid Micro-TSMD that was demonstrated at RADC. Westinghouse Micro-TSMD designs are presently incorporated and being tested on the F-16.

Rome Laboratory has determined that the integration of both of these technologies, Smart BIT and TSMD (or Smart BIT/TSMD), will enhance the detection and determination of false alarms.

## 1.2 OBJECTIVE

The purpose of Smart BIT/TSMD is to develop a test bed to demonstrate and assess the integration of the two technologies, Smart BIT and TSMD. Smart BIT provides the process for incorporating real-time environmental stress data as input to its reasoning process. TSMD research has emphasized the measurement, collection, storage, and failure correlation of stress data, with little attention to the real-time use of that capability. TSMD sensors convert data for analysis, data compression, and datalogging within the TSMD. The Smart BIT computer periodically interrogates both LRU BIT and TSMD data sets, using AI techniques to determine the UUT's true status, i.e., faults, false alarms, etc.

Smart BIT/TSMD provides Rome Laboratory with a laboratory test bed to evaluate and assess the individual characteristics as well as the integration of the two technologies. In this program, an emulated LRU (test vehicle) is modified by adding TSMD sensors (accelerometer and temperature sensors) and circuitry for injecting faults. The faults are injected into the test vehicle's BIT circuitry simultaneously with a corresponding change in temperature and/or acceleration to replicate false alarms and intermittent BIT failure reports.

## 1.3 APPROACH

In order to accomplish the above objectives, the following major tasks were achieved:

- A versatile test bed has been developed that allows for the study and integration of Smart BIT and TSMD technologies while retaining their independent capability
- Environmental stresses can be simulated or induced by actual sensors or combined for study
- The test bed is not limited to a single avionic LRU but is capable of simulating any 1553B bus-related LRU
- Software modifications can be made without external support. All development tools reside on the test bed
- The test bed simulates real-world scenarios, derived from actual data captured from in-flight sensing
- The test bed provides for the use of new field data as it becomes available.

## 1.4 SYSTEM OVERVIEW

The test bed consists of four computers: UUT computer, LRU/BIT computer, TSMD computer and the Smart BIT computer. A block diagram of the test bed is shown in Figure 1 (see Page v).

The UUT computer is used to control the test bed scenario execution, simulate the accelerometer and temperature sensors, and simulate LRU BIT behavior. It can provide preprogrammed environmental data in the absence of environmental chambers and vibration tables which will simulate environmental conditions. It also directs the LRU BIT computer how to respond to MIL-STD-1553B bus data queries it receives in order to simulate the LRU. Lastly, this computer has tools to generate the data bases necessary to perform the above mentioned tasks.

The LRU/BIT computer is used to simulate the LRU of the system. It generates "realistic" MIL-STD-1553B bus traffic in response to the Smart BIT computer's queries. This computer chassis also holds the TSMD sensor card which includes the environmental sensors and signal conditioning circuitry. The sensors are remotely locatable so that either the simulator or an actual LRU may be subjected to environmental conditions created by vibration tables and environmental chambers. The UUT computer may replace these sensors and simulate the signal conditioning circuitry when environmental simulation equipment is not available.

The TSMD computer simulates the TSMD and processing of environmental data. The conditioned output received from the sensor card of the LRU BIT computer is sampled and TSMD processing is performed on the analog data for storage. Additionally, the TSMD computer provides this environmental data to the Smart BIT computer for use in filtering out "false alarm" failures.

The Smart BIT computer receives the MIL-STD-1553B bus data from the LRU or LRU simulator and environmental data from the TSMD computer and operates on those data to filter out false alarms.

## 1.5 ORGANIZATION OF THE REPORT

The remainder of this report is organized as follows:

- Section 2 discusses the integration factors required to develop a test bed indicative of a real-world integrated system
- Section 3 discusses the individual hardware and software details of the test bed

- Section 4 discusses the integration criteria that accomplished the actual integration of the two technologies, Smart BIT and TSMD
- Section 5 discusses the demonstration scenarios for study of the effects of the integration
- Section 6 discusses the mechanism for installing an actual UUT in lieu of the test vehicle provided as part of the test bed for simulating any LRU
- Section 7 discusses the unique challenges that the development of the test bed provided
- The final section (8) states the conclusions and recommendations that can be drawn from the results of the study and demonstration.

# 2 – INTEGRATION FACTORS

In order to integrate Smart BIT and TSMD, various items are required. Obviously Smart BIT and TSMD techniques are among them. In addition, viable data, on which both Smart BIT and TSMD are to operate, are essential. Otherwise, the entire basis of their integration would be meaningless. These data are used to build real-world scenarios for injecting faults into an LRU simultaneously with temperature and/or acceleration. As a result, false alarms/intermittent BIT failure reports can be studied along with environmental changes. The final ingredient is a suitable LRU on which to accomplish the above integration factors.

## 2.1 AIRCRAFT ENVIRONMENTAL FACTORS

### 2.1.1 Rationale and Basic Equations for Vibration

One of the original goals of the Smart BIT/TSMD project was to use actual aircraft environmental data recorded in flight. Towards that end, we requested flight data on the ALQ-131 from an on-going program at Warner Robins. However, due to delays in obtaining those data, we had to defer to using older data recorded on the F/FB-111, and documented in Reference 7.

Because of time and budget constraints of our Smart BIT/TSMD Integration project, we restricted our environmental scenarios to temperature and vibration effects. This kept us compatible with the existing Smart BIT-2 software, minimizing software changes. Since vibration is by far the more complex of the two parameters, it is discussed here at length.

In order to determine a unit's sensitivity to vibration, vibration must be sensed in three axes. For simplicity, we considered a single-axis accelerometer simulation and assumed that it is oriented in the sensitive direction. The actual flight data in Reference 7 are structural vibrations for the Tail Pod assembly. Reference 7 was chosen because it gave vibration data for many different modes of flight, including level flight, weapons drop, level turn (constant g and maximum g), speed brakes out, and side slip. All the surveys for these various modes recorded the processed raw data as spectral density in $G^2/Hz$ versus frequency in Hz. The worst-case scenario occurred for the speed brakes out mode and is shown in Figure 3.

processed raw data as spectral density in $G^2$/Hz versus frequency in Hz. The worst-case scenario occurred for the speed brakes out mode and is shown in Figure 3.

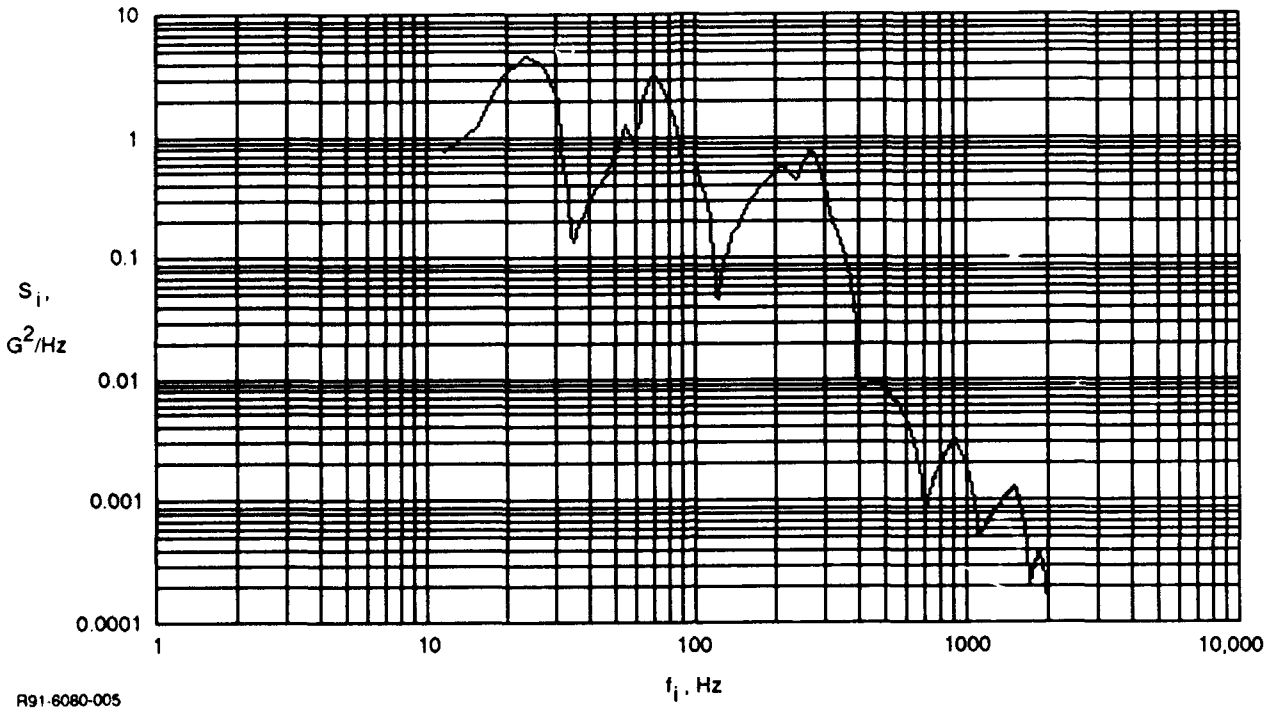



**Figure 3. Structural vibration spectral density, speed brakes out.**

Having flight data in the frequency domain (spectral density) leads to a problem since the UUT computer must simulate vibration using timeline data, i.e., g's vs time. In addition, we must simulate the signal using a digital computer and a D/A converter, requiring the signal to be discretized.

The following subsections discuss the methodology used to solve the above problem. However, before proceeding with the discussion, the following basic equations are given as a reference. In the continuous frequency and time domain, the equations are defined as follows:

$$\text{Spectral Density} = S(f)$$

Here,

$$S(f) = \lim_{T \to \infty} (2/T) \cdot |G_T(if)|^2 \qquad (1)$$

where

$G_T(if)$ is the Fourier transform of g(t)
over the finite interval T
f = frequency
$i = \sqrt{-1}$

$$g_{rms} = \sqrt{\int_0^{\infty} S(f)df} \tag{2}$$

In the discrete frequency and time domain, the equations become

$$S(i) = |G(i)|^2/\delta f \tag{3}$$

and

$$g_{rms} = \sqrt{\sum_i S(i) \cdot \delta f} \tag{4}$$

where

i = index for frequency domain
$\delta f$ = incremental change in frequency

$$g_{mean} = G(0) \tag{5}$$

In addition, since g(j) is random with a normal distribution, if we wish to calculate the rms and mean values of g from the time domain, the equations would be:

$$g_{mean} = \text{Mean}(g) = \Sigma\, g(j)/N \tag{6}$$

$$g_{rms} = \text{Standard Deviation (g)} = \sqrt{\sum_j (g(j) - g_{mean})^2/N} \tag{7}$$

where

j = index for the time domain
N = number of discrete points in g
g(j) = discrete vibration time function.

Thus, we can calculate the rms value of g from the spectral density (Equation 4) or the time function (Equation 7). We will also be able to calculate the mean value of g from frequency domain results (Equation 5) and directly from g in the time domain (Equation 6). Calculating the mean ($\mu$) and standard deviation ($\sigma$) in both domains, the validity of the transformation from the frequency to the time domain can be checked.

### 2.1.2 Graphical Data

The basic data given in Reference 7 are graphical plots of spectral density (S(f)). We manually read the S(f) and frequency breakpoints from the graphs and entered them into our transformation program. The breakpoints were input to a linear interpolation function to create the discrete S(i) data in the frequency domain, where frequency ranged from 0 to 2000 Hz. A maximum index of Ni = 512 was defined, resulting in $\delta f = 3.91$ Hz. These numbers are basically derived from the Nyquist criterion in the time domain.

Having S(i), the aircraft structural vibration spectral density, the next step was to derive the Printed Circuit Board (PCB) vibration spectral density. References 8 and 9 discuss vibration data with sensors at various locations in the aircraft, including aircraft structure, equipment bay, electronic box, electronic subassembly, and circuit board. Comparing data from these references, it is apparent that transmission of the vibration from structure to a circuit card varies widely depending on the electronic box in question. Since the F-111 Tail Pod data were reduced from structural sensor data only, we have to assume a spectral density attenuation factor to derive the circuit board S(i). For this project, we assumed that the board resonances have been designed out. We also assumed an attenuation factor of 1% in the low-frequency (10 to 960 Hz) band, 10% attenuation in the high (1040 to 2000 Hz) band, and a linear transition between bands. Multiplying the reference S(i) by these factors results in the typical vibration spectral density shown in Figure 4, representative of the vibration at the circuit board level.

### 2.1.3 Deriving Time Domain Data

From Equation 3, it is possible to derive the magnitude of the Fast Fourier Transform (FFT) of g(j), i.e., |G(i)|:

$$|G(i)| = \sqrt{S(i) \cdot \delta f}$$

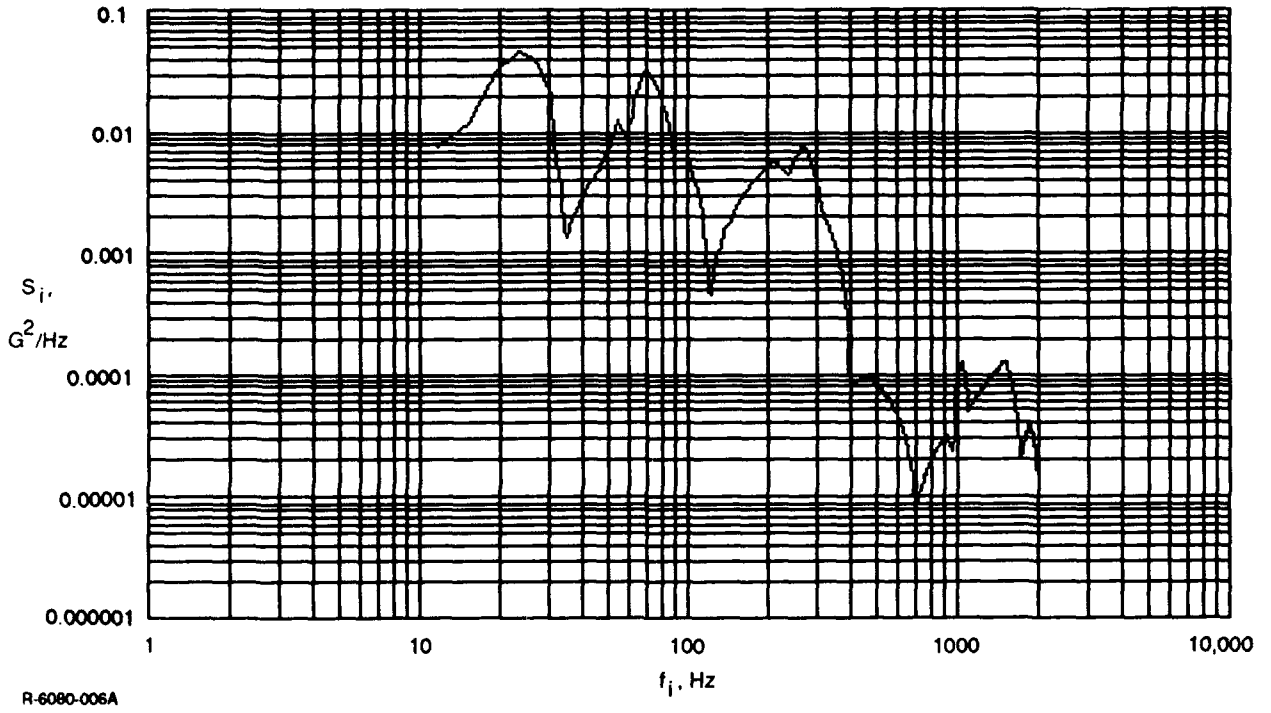



**Figure 4. PCB vibration spectral density, speed brakes out.**

However, we have no way of knowing the phase angles for these magnitudes. We therefore assumed that the phase is a random variable between 0 and $2\pi$ with a uniform distribution. This is reasonable since it results in a distribution for g(j) which is very close to a normal distribution. This assumption gives us a complete FFT.

Having the above results, we are able to use the Inverse Fast Fourier Transform (IFFT) formula:

$$g(j) = \Sigma\ W(i) \cdot e^{-2\pi\sqrt{-1}\ j\ i/N}$$

where

$W(i) = G(i), G^*(N-i)$

That is, we take the complex conjugate of G(i) in reverse order and append it to G(i), forming the vector W(i)

N is the dimension of W(i) and g(j)

j is the time index

i is the frequency index.

This results in the time function g(j), shown in Figure 5.

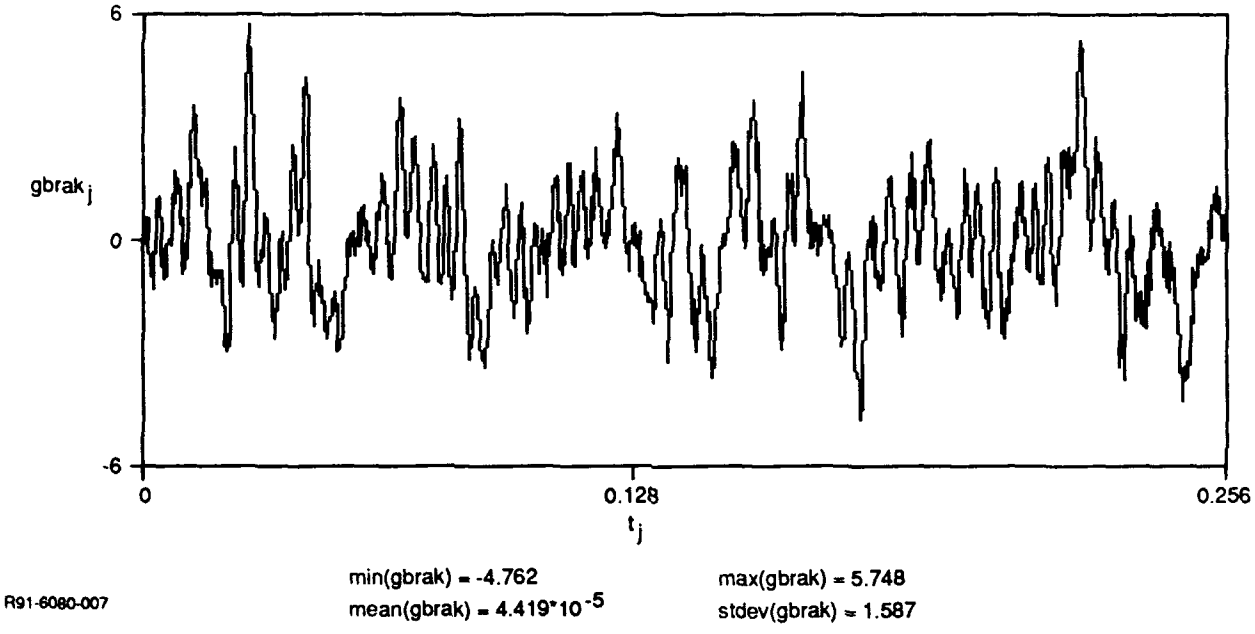



**Figure 5. PCB vibration time line, speed brakes out.**

When we first used the IFFT to derive the time function, we were considerably off in our magnitudes. The problem stemmed from the way our particular software package normalized the IFFT and FFT functions. In order to assure proper magnitudes, we calibrated our program using band-limited white noise with $S(f)=S_0=1$ for the pass band. Band-limited white noise is a function for which Equation 1 converges by definition, and for which Equation 2 results in a closed-form equation for calculating the rms value of the continuous noise function as follows:

$$g_{rms} = \sqrt{S_o \bullet \text{Bandwidth}}$$

Knowing this, we determined and verified the constants required for our particular normalization. Successive transformations using IFFT and FFT were made with band-limited white noise spectral density as input. The $g_{rms}$ values for the three cases compared to the closed-form calculation as follows:

$g_{rms} = 44.721$ (reference, closed form equation)
$g_{rms1} = 44.765$ (S(i) and Equation 4)
$g_{rms2} = 44.699$ (g(j) and Equation 7)
$g_{rms3} = 44.764$ (IS(i) and Equation 4)

The $g_{mean}$ values compared to each other as follows:

$g_{mean1} = 1.398$ (G(i) and Equation 5)
$g_{mean2} = 1.398$ (g(j) and Equation 6)
$g_{mean3} = 1.398$ (IG(i) and Equation 5)

The computer run for the above values is given in Appendix A. As can be seen here, the results compare very well.

### 2.1.4 Conversion to D/A Data

As shown in Figure 5, the vibration signal time interval (TI) is approximately 250 msec. This is a result of the resolution in the time domain (Nj = 1023) and the Nyquist criterion ($\delta t$ = 0.25 msec), since TI = Nj•$\delta t$. Since the simulation required a 1-sec time interval, one option was to increase Nj. However, that resulted in exceeding memory capacity in our computer run. Thus, in order to get the full second of time required, we appended three g(j) time series to the original series. While this is a pseudo random signal, it was sufficient for our purposes.

The final step was to scale the 1-sec vibration signal for the particular D/A converter used in the UUT computer. The equations defining this conversion were as follows:

$$Ndac = 4095$$

$$gdac(k) = \{[g(k) + 10]/20\} \bullet Ndac$$

$$gdac(k) = if\{[gdac(k) - floor(gdac(k)] > 0.5, ceiling[gdac(k)], floor[gdac(k)]\}$$

where

k = the new time index.

This resulted in integer values in the range from 0 to 4095, where

$g = -10 \Rightarrow gdac = 0$
$g = 0 \Rightarrow gdac = 2047$
$g = 10 \Rightarrow gdac = 4095$

### 2.1.5 Flight Modes

In addition to the above transformations and conversions done for the speed brakes out flight mode, the whole procedure was repeated for the remaining flight modes, i.e., normal level flight, level turn constant g, level turn max g, weapons drop, and side slip. Appendix B contains the PCB spectral density curves and the resulting vibration time lines for all flight modes for reference.

### 2.1.6 Correlation of BIT Faults to Vibration

Having derived the timeline data for the vibration signal, the next question to be resolved is how BIT faults should be correlated with the occurrence of vibration. At the time of this project, we did not have actual aircraft data of BIT versus vibration to analyze. As a result, we had to resort to expert opinion to determine with which vibration parameter ($g_{rms}$, $g_{peak}$, $\int g\,dt$) to correlate the occurrence of faults. We conferred with several experts at Grumman. Some believed that vibration problems are designed out during development and are not a problem once the aircraft is in the field. However, if problems do occur, the consensus was that vibration peak is a good parameter with which to correlate faults. This opinion was corroborated by Reference 10, which actually derived an equation for predicting component failure on a PCB based on the amount of time spent exceeding a peak value limit of vibration. As a result, we correlated BIT failures with peak values of vibration exceeding a given limit. The experts agreed that these failures could be intermittent or hard in nature.

### 2.1.7 Temperature and Vibration Interaction

We also discussed temperature effects with some of the above experts and in addition, with a thermal expert. One of the interesting points that surfaced was that temperature can interact with vibration; it is possible for the two to cause similar effects on a board. Thus, although each may be within limits, their combined effect causes failures. The original Smart BIT-2 software could not handle this problem. In the course of the Smart BIT/TSMD project, changes were made to the neural net learning supervisor which did improve software performance. However, this is only a partial solution to the problem and further work needs to be done.

### 2.1.8 Temperature

Considering temperature by itself, it is possible under anomalous cooling conditions for temperature to cycle out of specification and cause a failure. Here again, we do not have BIT versus temperature flight data and we are relying on the opinion of the experts. The consensus

agreed that not only are high and low temperature values important, but rate of change of temperature is also significant.

Other important factors include box location in the aircraft. Three types of locations are usually considered:

- Non-conditioned bay (-65° to 270°F)
- Conditioned bay (-65° to 160°F)
- Cockpit (60° to 80°F).

Types of boxes also vary. Some are forced-air-cooled and others are ambient-cooled.

In general, there are two types of failures that can be induced by temperature: mechanical fatigue on component connections and electrical parameter changes in components. The mechanical fatigue case is the one we consider additive to vibration effects.

We have included scenarios where temperature either exceeds a high limit, but not critically high, and cycles at a rate considered reasonable for temperature. BIT faults were made to occur at a more frequent rate for faster temperature cycles.

### 2.1.9 Uncorrelated Versus Correlated Faults

The problem referred to here relates to the occurrence of BIT fault indications which are not temperature or vibration induced, occurring along with BIT fault indications that are caused by vibration or temperature. This is shown in Figure 6. As can be seen, vibration has subsided during time interval T but BIT fault indications occur several times.

In order to address this problem, we proposed changes to the temporal monitoring. However, development of that solution had to be abandoned because of the more basic and time-consuming porting challenges described in Section 7. In addition, solving this problem was beyond the contract scope.

## 2.2 SMART BIT AI TECHNIQUES

There are four Smart BIT AI techniques that are used by the Smart BIT computer for analyzing the TSMD and LRU BIT data to determine intermittents, false alarms, and hard fault LRU conditions:

- Information Enhanced BIT
- Improved Decision BIT
- Adaptive BIT
- Temporal Monitoring.

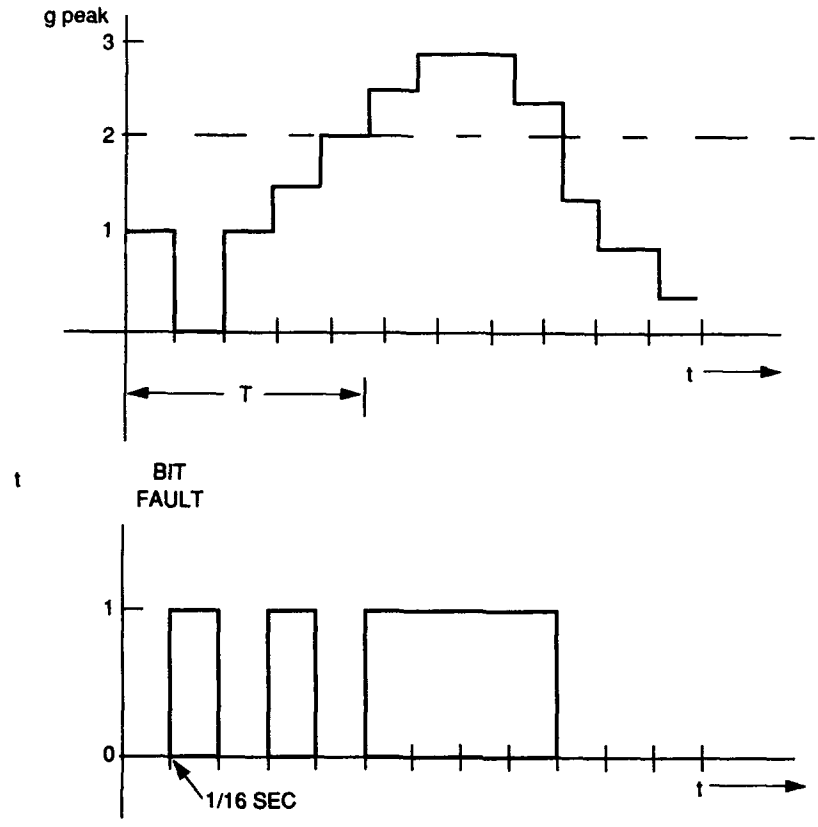



**Figure 6. Correlated vs uncorrelated faults.**

Information Enhanced BIT is a technique that uses additional sensory data to complement the standard BIT information. Sensory information such as power supply status, temperature, and vibration are analyzed together with the BIT data to determine the relative health of an avionics system.

The Improved Decision BIT technique uses Temporal Monitoring to overlook transient conditions and other short-term anomalies by making decisions about BIT behavior based on past "experience" rather than on a constant fault rate that was determined in the laboratory under static test conditions.

Adaptive BIT is a method that attempts to classify the status of an avionics system by associating the behaviors it is currently observing with similar behaviors that it has seen in the past. Adaptive BIT "adapts" or tries to "learn" the conditions when a failure has taken place (e.g., Did

this fault occur while vibration was high?). Two methods are provided in the Smart BIT/TSMD Integration System test bed to illustrate Adaptive BIT: Neural Networks and K-Nearest Neighbor.

Temporal Monitoring BIT combines a number of methods for determining the LRU fault status. Temporal Monitoring BIT tries to model the LRU BIT data over extended periods of time to correlate previous fault information with the one now being analyzed. This requires combining a Markov model finite-state machine of intermittent behavior with Bernoulli random variables. The Markov model finite-state machine consists of four possible states that reflect the relative health of the avionics system and are: OK (no fault), hard fault (consistently faulty behavior), intermittent faulty (fault conditions over a short period of time), and recovering (was faulty but possibly appears as if the behavior will return to a normal, no-fault condition shortly). The Temporal Monitoring BIT software determines the state that represents the present status of the BIT behavior (according to what it has previously observed over time) based on state transition probabilities. These probabilities are dynamically adjusted according to the behavior of the LRU BIT reports thus far.

Therefore, Improved Decision BIT and Information Enhanced BIT are general methods for making better BIT evaluation decisions. Adaptive BIT and Temporal Monitoring techniques are refinements of Improved Decision BIT and Information Enhanced BIT. Hence, discussions in this report that refer to Adaptive BIT and Temporal Monitoring techniques are also referring to Improved Decision BIT and Information Enhanced BIT but in a more focused and specific way. All of these techniques were derived from the Smart BIT-2 contract (Contract No. F30602-86-C-0252) and have been expanded for the Smart BIT/TSMD Integration System test bed.

The following summarizes the specific Adaptive BIT techniques used in the Smart BIT/ TSMD Integration System test bed (a more detailed discussion of both Adaptive BIT and Temporal Monitoring techniques is located in Appendix E, excerpted from Ref. 3).

Neural Networks are based on a technology that diverges substantially from the standard Von Neumann machine model for computing. Neural Networks do not follow a sequential software program as in traditional methods for Von Neumann machines. Instead, information about a subject is contained in the weights of connections (relative strengths) between neurons (multiply-accumulator processing elements) as well as in the mechanism of interconnecting the neurons.

The information is adaptively "learned" (weights are adjusted) by the Neural Network as determined by the selected learning law. Neural Networks are highly adaptive feature extractors that cull out the distinguishing "trademarks" of the input data set, thus allowing later "recognition" when presented with similar data. They are massively parallel, highly interconnected and therefore potentially very fast (when implemented in actual hardware), and quite robust (if a small proportion of neurons are damaged or the weights are damaged, or the input data has noise, performance will not degrade precipitously but instead will degrade gracefully). The Neural Network paradigm used in the Smart BIT/TSMD Integration System Adaptive BIT technique is called Back Propagation. The Back Propagation's learning law is actually a modified version of the Least Mean Squares Delta Rule (not biologically based) by Widrow and Hoff. The Adaptive BIT Neural Network attempts to "learn" the associations of the BIT status and the environmental sensor data (temperature, etc) so that it might classify the LRU status more quickly and accurately.

The K-Nearest Neighbor Adaptive BIT method is a pattern-matching algorithm that attempts to classify the avionics status by comparing the present status behavior to previous status patterns it has saved in its memory (status behavior will include the BIT status as well as any sensory data). The K factor has been chosen to be five (this is a programmable value in the Smart BIT Parameters Menu Window) because this value yields the best overall performance. The K factor determines how many "nearest neighbors" with similar behaviors there must be for the avionics status behavior presently being observed to be classified with that group. For instance, if the present status behavior is closest to four other status behaviors that have been observed and classified together previously, then it will be classified with that group. If not, the present behavior will be classified in its own group. K-Nearest Neighbor is adaptive in the sense that it improves its classification accuracy as more status behavior samples are collected and thus larger numbers of behaviors are clustered into similar behavior groups.

## 2.3 TSMD TECHNIQUES

Time Stress Measurement Devices (TSMD) have been developed to provide a way to measure environmental characteristics (such as temperature, vibration, and shock) to gain additional information about the operational state of a system during its normal operation for maintenance purposes. One of the primary maintenance criteria for acquiring this environmental data is to discover whether some equipment failures and intermittents may be related to environmental characteristics. See Appendix H for a detailed discussion of prior TSMD research.

There are many implementations possible for capturing environmental data and discerning environmental events from that data. Likewise, several approaches are possible for distilling the processed data such that the data may be accumulated within the TSMD itself and saved for later examination and analysis when it is convenient to do so. Also, prime mission equipment design considerations place severe constraints on the TSMD implementation, such as minimal size and low power. All of these design constraints force trades between capabilities, resources, and processing that can be used for an effective TSMD implementation. Specific interface requirements for the TSMD implementation in the Smart BIT/TSMD Integration test bed are shown in Figure 7. These interface requirements are defined in the Interface Design Document for this effort.

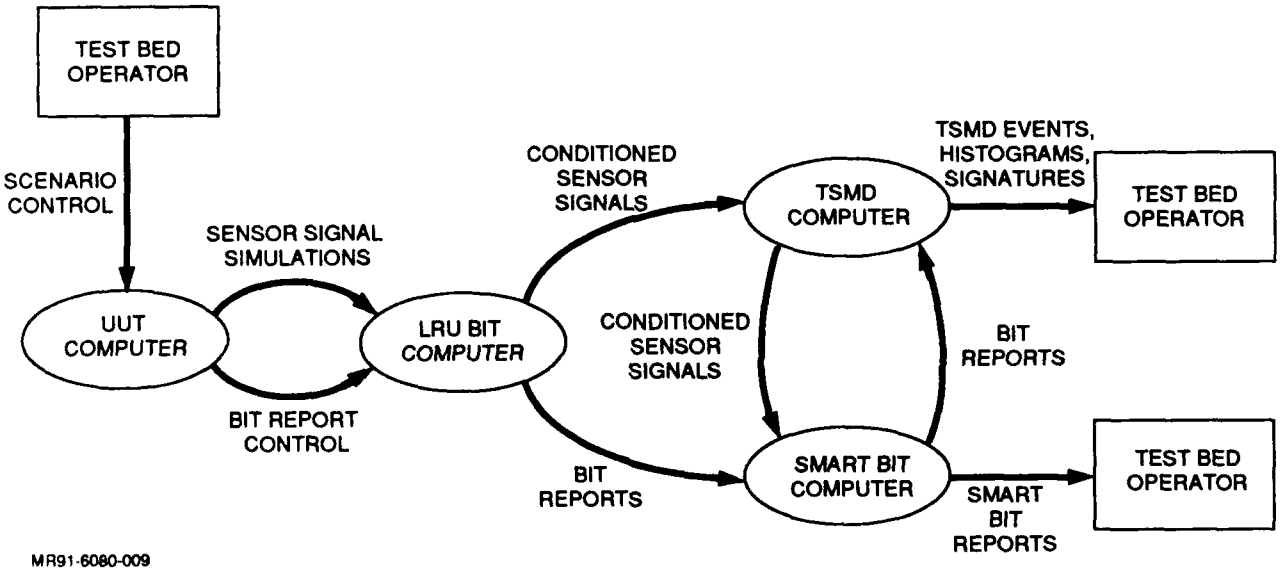



**Figure 7. Smart BIT/TSMD Integration System data flow diagram.**

The TSMD circuitry and techniques for the Smart BIT/TSMD integration system have been carefully designed to provide the same capabilities as a fielded TSMD in an avionic environment. This results in an implementation which does not fully utilize the power and resources of the personal computer, but does provide what can be referred to as a "field-ready" configuration. An example of this would be the storage of TSMD event data that is structured and compressed as if it resided in a nonvolatile memory component, such as an EEPROM integrated circuit, rather than a much larger, more flexible structure allowable within the PC computer memory architecture. By adhering to the TSMD implementation constraints of a fieldable device, the software assessed and evaluated on the PC computer can emulate and be directly transferred to actual TSMD hardware for installation in an operational, fielded avionic system.

The TSMD techniques selected for the Smart BIT/TSMD Integration System uses analog circuitry to discriminate the sensor signals for environmental characteristics of interest (such as extracting vibration from an accelerometer) rather than digital signal-processing techniques employing Fast Fourier Transform (FFT) calculations. A microprocessor is used to measure the conditioned sensor signals with an 8-bit analog-to-digital (A/D) converter. The samples are then processed using the following algorithms:

- Life-stress monitoring
- Over-stress event logging
- Fault event logging.

These algorithms are described in detail in the following subsections.

### 2.3.1 Life-Stress Algorithm

Life-stress monitoring is a paradigm for collecting long-term sensor data. This function provides information for off-line reliability analysis of the equipment being instrumented by indicating the operational environment over the life of the system.

### 2.3.2 Over-Stress Algorithm

Over-stress event logging provides short-term sensor data useful for determining the environmental profile to which the instrumented system has been exposed. Since time-correlated sensor data are typically collected at a rate of one sample per second or faster, this function's capacity is constrained by the short-term memory size. Since design goals for the TSMD memory are small (less than 100 Kbytes), techniques such as thresholding and data compaction are employed. Thresholding uses a series of preprogrammed values to establish ranges of interest, and events are generated when the sampled sensor data moves from one range to another. These events are time-stamped and stored in nonvolatile memory for off-line analysis. Data compaction techniques are used to maximize the available memory and optimize the event log storage.

### 2.3.3 Fault Logging Algorithm

Fault logging records the environmental characteristics (temperature, vibration, voltage, and shock) during the occurrence of a fault. Though not specifically required by the Smart BIT/ TSMD Integration SOW, this capability provides the best operational data for off-line failure

mode analysis since the environmental characteristics are directly correlated with the occurrence of a fault. These data would be a primary source for historical trending and fault-stress correlation analysis for establishing information used in the Smart BIT paradigms, as well as a basis for establishing and fine-tuning sensor data ranges used in TSMD thresholding functions.

Fault logging is mechanized by collecting and storing sensor data samples prior to, during, and after the occurrence of the fault in nonvolatile memory for off-line analysis. In addition to a time-stamp, the Maintenance Fault List (MFL) code would be stored along with the sensor data in the nonvolatile memory. The short-term memory is a cyclical buffer and holds the most recent samples of sensor data. When a fault is indicated, additional samples are collected and the entire cyclical buffer is stored in long-term memory along with the time/date stamp and BIT fault code. This implementation provides a fault signature with the fault occurrence centered within the signature. The cyclical buffer length and the number of samples recorded after the fault indication are software programmable to facilitate assessment of the capacities and effectiveness of the fault signature feature. Overlapping fault indication is handled by storing consecutive cyclical buffers.

## 2.4 EQUIPMENT SELECTION

The avionics system selected for simulating the LRU equipment was the Standard Central Air Data Computer (SCADC). This decision was based on the fact that the SCADC represents a typical modern avionic system that is used on a wide variety of Air Force aircraft. In addition, information about the SCADC was already available from work done on the Smart BIT-2 contract (Contract No. F30602-86-C-0272). An in-depth discussion of the SCADC system and its BIT is provided in Appendix G of the Design Plan for this effort.

## 3 – SMART BIT/TSMD INTEGRATION TEST BED

The Smart BIT/TSMD Integration System provides a test bed for investigating various Artificial Intelligence (AI) and Time Stress Measurement Device (TSMD) techniques for the purpose of evaluating their usefulness in reducing the number of Retested OK (RETOK) and Cannot Duplicates of avionic system Line Replaceable Units (LRU). The Smart BIT/TSMD Integration System employs information gathered from the LRU's Built-In-Test (BIT) as well as environmental factors from TSMDs in order to determine what state the LRU is in (hard fault, OK, intermittent, etc). It consists of four computers: the UUT computer, the LRU BIT computer, the TSMD computer and the Smart BIT computer (see Figure 8 for the Smart BIT/TSMD Integration System overview block diagram). The UUT computer provides the Fault/OK profile and parametric information to the LRU BIT computer so that it can perform a partial simulation of a SCADC LRU. During a scenario, the UUT computer commands the LRU BIT computer to

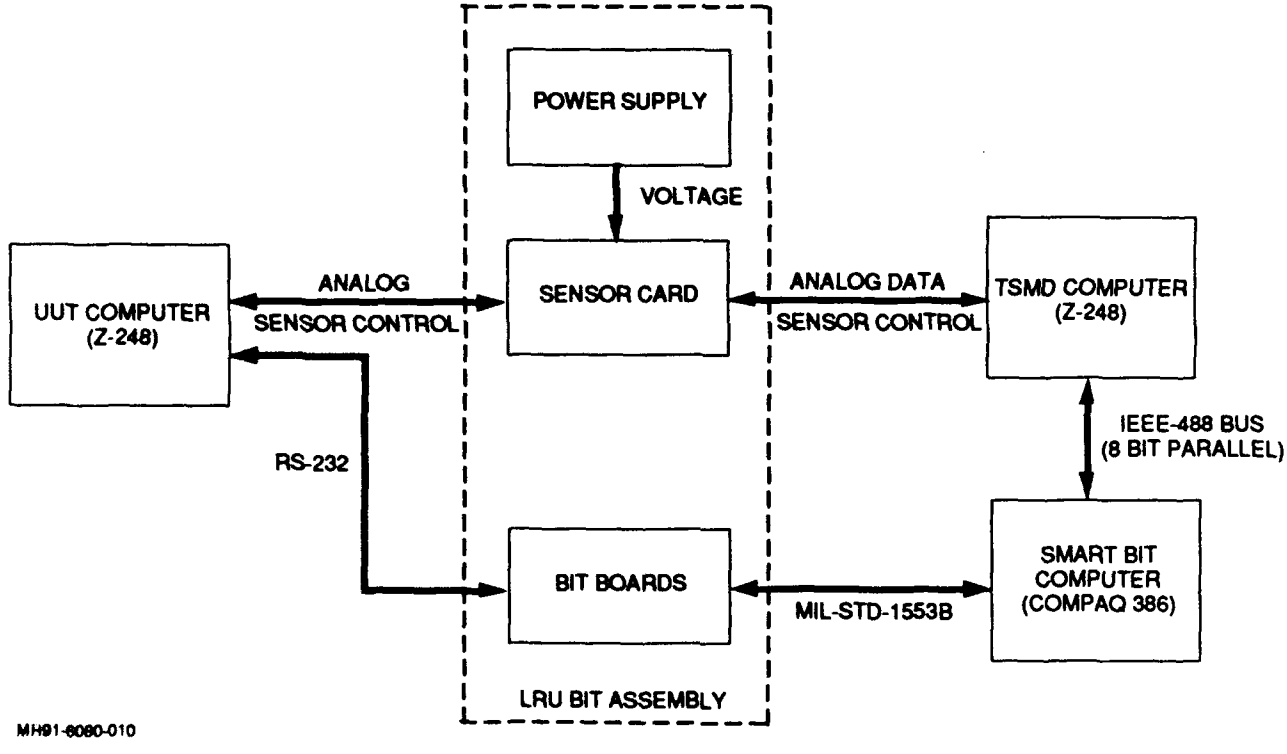



**Figure 8. Smart BIT/TSMD Integration System**

change its simulated SCADC MIL-STD-1553B message responses to the Smart BIT computer to be either OK or contain BIT fault information. The UUT computer also provides a simulation of sensor data (temperature and accelerometer) and sends these signals to a TSMD sensor card (resident in the LRU BIT computer chassis) which conditions this analog data (actual temperature and accelerometer sensors may be selected instead of a UUT computer simulation). The TSMD computer reads the conditioned sensor data and reports this information whenever a periodic request from the Smart BIT computer occurs. The Smart BIT computer combines the TSMD computer environmental data with the LRU BIT computer messages and analyzes them to determine the relative health of the LRU.

## 3.1 UUT COMPUTER

The UUT computer is a Zenith Z-248 IBM PC/AT compatible microcomputer. Its function is to provide software tools to create timeline sensor and BIT profiles as well as LRU scenario data. It also sets up the LRU BIT computer with the scenario data for a typical SCADC (Standard Central Air Data Computer) LRU. It runs a scenario by using the timeline sensor and BIT profiles to control (using a multifunction card contained within the UUT computer) the TSMD sensor card analog data (conditioned signals from the sensor card are reported to the TSMD computer) as well as the digital LRU BIT reports (Faulty or OK) to the Smart BIT computer.

### 3.1.1 UUT Computer Hardware

The UUT computer hardware consists of a 12-MHz Zenith Z-248 microcomputer with 1.1 Mbytes of RAM, a hard disk and floppy drive, a monochrome monitor, a mouse, and a COTS (Commercial-Off-The-Shelf) multifunction card containing analog and digital I/O, as well as onboard timers and interrupt capability. All hardware in the UUT computer are COTS equipment and therefore no hardware design work was required for the UUT computer subsystem.

### 3.1.2 UUT Software

The UUT computer provides the LRU scenario, BIT, and TSMD time line editors to generate data files for stimulating the conditioning circuitry of the TSMD sensor card. In addition, the UUT computer controls the simulation of normal, false alarm, and hard fault BIT behaviors in the LRU BIT computer. The UUT computer also provides the controls to select whether sensor data (that is sent to the TSMD sensor card residing in the LRU BIT computer chassis) is simulated by the UUT computer or originates from the actual temperature and accelerometer sensors (or a combination of both). The UUT computer software is written in the the 'C' programming

language and makes use of the Microsoft Windows multitasking environment (with a mouse) to provide an easy user interface to generate the LRU scenario, LRU BIT, and TSMD data files.

The UUT computer contains analog I/O signals for sensor signal output simulation and discrete output signals for TSMD sensor selection control, as well as an RS-232C serial digital bus for LRU BIT computer scenario downloading and control. The UUT computer has two digital-to-analog output signals that provide a simulation of temperature and acceleration sensor signals to the TSMD sensor card (residing in the LRU BIT computer). It also contains two analog-to-digital inputs to wrap the analog outputs back for testing purposes. The discrete output signals are also sent to the sensor card to control the selection of either the UUT analog outputs or actual temperature and accelerometer sensor outputs, or a combination of both. In addition, a discrete output controls a signal for determining when a simulated voltage is applied to the LRU BIT computer. (This signal is simulated because power is always applied to the LRU BIT computer. Its purpose is to indicate to the Smart BIT and TSMD computers when the scenario should begin and end.) Lastly, the RS-232C interface is used by the UUT computer to send LRU parametric and BIT information to the LRU BIT computer. The UUT computer also uses the RS-232C serial interface to control the LRU BIT computer's BIT behavior (Faulty or OK) during the time a scenario is running.

The UUT computer software design (see Figure 9) consists of a Main Program, a Window User Interface, a Scenario Generation Tool, a Sensor Data and BIT Time Line Editor, a Run Scenario Executive, and Interface Drivers for the hardware.

**3.1.2.1 Main Program** – The UUT computer software sets up the Main UUT Window, Time Line Window, and menu bar titles for the Microsoft Windows environment. It creates and displays just the Main UUT Window initially. It then starts the Main Microsoft Windows Message Read/Translate/Send Message Processor loop. This loop receives the Microsoft Windows keyboard, mouse position, mouse button, timer interrupt, and windows information messages from the Microsoft Windows message queue. It translates them into a common format and dispatches them to a procedure that processes each message type.

**3.1.2.2 Window User Interface** – The Window User Interface registers the Main UUT Window and the Time Line Window object classes with the Microsoft Windows environment. The

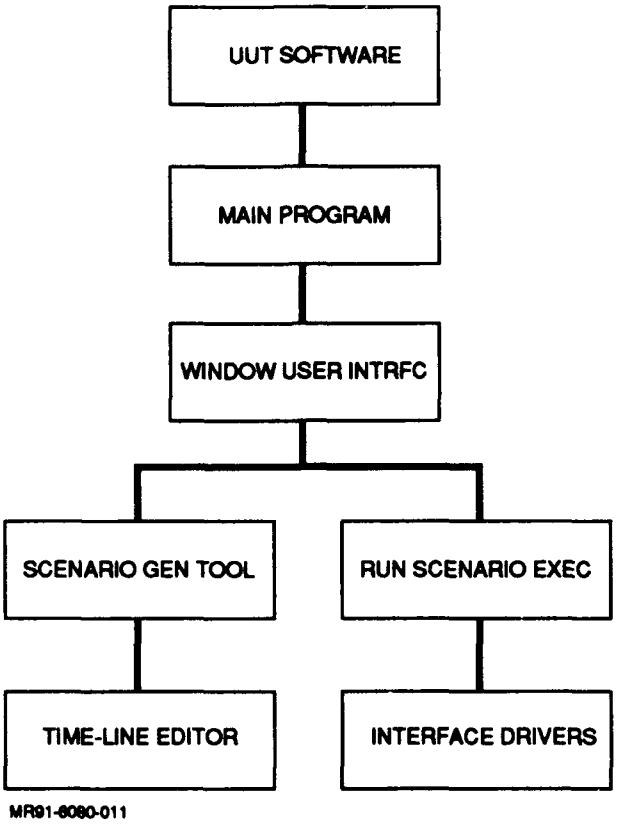



**Figure 9. UUT software top-level block diagram.**

Window User Interface then sets up and opens the UUT to LRU BIT computer RS-232C serial (second serial port) channel, and loads a data file for creating seven "canned" accelerometer waveforms. The Window User Interface also contains programs for opening, reading, copying, deleting, and saving various control and data files that are required to set up and run scenarios.

**3.1.2.3 Scenario Generation Tool** – Selecting the UUT Scenario Generation Tool creates (from the Tool menu) a set of windows and dialog boxes using calls to Microsoft Windows for creating and editing a scenario file. A scenario file contains the parametric information data format that an actual avionics LRU would transmit over the MIL-STD-1553B bus (see Table 3 for an example of a typical scenario file). Information such as the LRU BIT word, operational data such as altitude (for a Standard Central Air Data Computer), and how the data changes during a fault condition are described in the scenario data file. The purpose of the Scenario Generation Tool is therefore to act as a text editor tool for the creation, review, and editing of the scenario data file.

**Table 3. Scenario No. 1 (SCEN1. DAT File)**

; SCENARIO NAME: ALL OUTPUTS FIXED

;

; NEW MESSAGE:

; COMMAND IS SCADC MAIN OUTPUT MESSAGE:

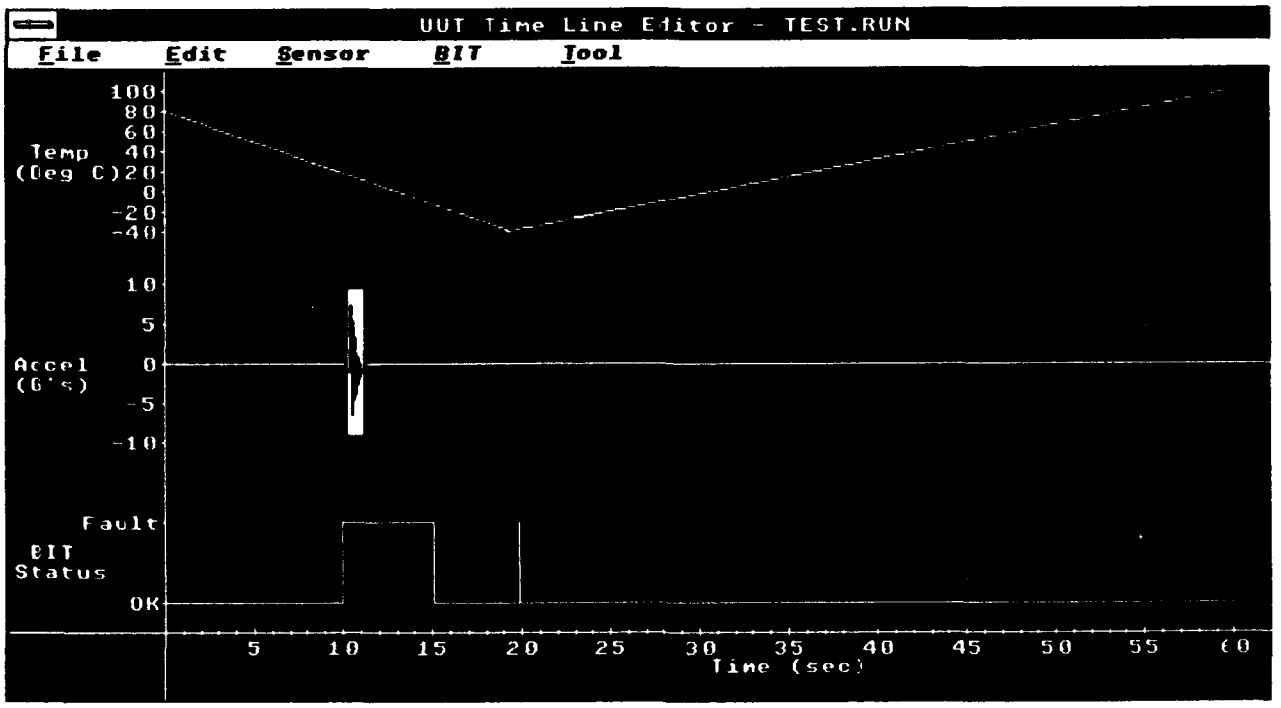
; RT # RT SUBADDRESS
28 16

; MODE CODE # MODE CODE DATA
99 99

| ; WORD # | DESCRIPTION | UNITS | DATA TYPE | INITIAL VALUE | INCREMENT VALUE | FAULT VALUE | FINAL VALUE | RESOLUTION | LSB BIT # |
|---|---|---|---|---|---|---|---|---|---|
| 1 | BIT_WD | NONE | BOOLEAN | 65280.0 | 0.0 | 31488.0 | 65280.0 | 1.0 | 16.0 |
| 2 | PRES_ALT | FT | SIGNED | 12200.0 | 10.0 | 0.0 | 14240.0 | 2.5 | 16.0 |
| 3 | BARO_ALT | FT | SIGNED | 12200.0 | 10.0 | 12200.0 | 14240.0 | 2.5 | 16.0 |
| 4 | TRUE_AIRSPEED | KNOTS | SIGNED | 350.0 | 2.0 | 350.0 | 540.0 | 0.125 | 15.0 |

MR91-6080-031

**3.1.2.4 Time Line Editor** – The Time Line Editor allows the user to synchronize the simulated sensor output with the selected BIT scenario. The Time Line Editor is actually three graphical editors displayed simultaneously in one window (temperature, accelerometer, and BIT graphical editors). All three editors are plotted against the same horizontal x axis (time) so that the data may be synchronized in time with each other (see Figure 10 for a typical UUT computer Time Line Editor display). *Selecting the Time Line Editor from the* Tool menu causes the UUT computer software to create and display the Time Line Editor Window along with its menu controls. Each graphic editor has its own set of menu edit controls for manipulating data as well as some shared controls for file I/O and erasing previous graphic data. Only one graphic editor may be active (have focus) at a time (thereby avoiding entry errors). The graphical editor that is currently active is indicated by a highlighted rectangular region (shown as a shade of amber against a black background on an amber monochrome screen) that the selected editor controls. When the operator selects a different editor, the focus is shown to be changed by removing the previously highlighted rectangular area and moving it to the newly selected editor's region. The temperature (top) editor is selected by default when the Time Line Editor is first entered. Selecting the accelerometer tool as the active editor is accomplished by using the mouse to select the Sensor menu and then choosing the desired accelerometer icon. The focus is then automatically moved to the accelerometer editor (focus is removed from the temperature editor first) and accelerometer waveforms may be placed anywhere in the accelerometer editor region using the mouse. Similarly, selecting one of the two options (OK, Fault) from the BIT menu results in the focus being changed to the BIT Status region.




R91-6060-012

**Figure 10. UUT time line editor display.**

The temperature Time Line Editor allows the simulated output of time-varying temperature from -40°C +100°C over a one-minute interval. The operator can create a unique temperature profile by repeatedly clicking and holding down the left mouse button and dragging the mouse to the right in the temperature editor region. The temperature Time Line Editor continually draws a line from the last line segment position to the cursor position while the left mouse button is depressed. Moving the mouse places the mouse cursor in a new position, causing the old line segment to be erased and a new line segment to be drawn to the present mouse cursor position. Releasing the left mouse button "fixes" the temperature line segment drawn to the last cursor position. The right end of this "fixed" line segment then becomes the start position of the next temperature line to be drawn to the cursor. Drawing small line segments for break point positions allows the linear approximation of a curve to "soften" the temperature change areas for a more realistic temperature profile. The option to read in a previously defined temperature profile is also easily accomplished by using the file I/O menu controls.

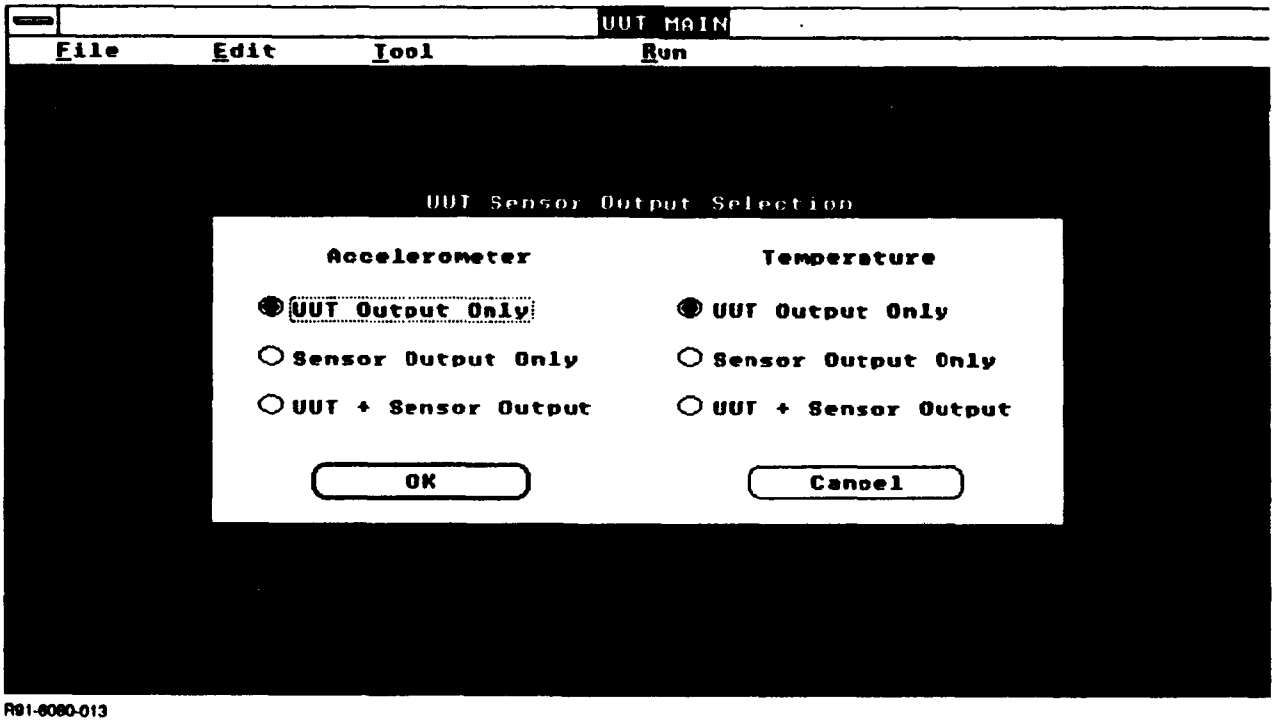
The accelerometer Time Line Editor allows the simulation of up to seven different accelerometer waveforms at selected times in the scenario within a one-minute interval. The Sensor menu contains seven unique icons that represent the accelerometer waveforms. These

accelerometer waveforms are based on actual flight data (see Subsection 2.1) that have been translated into the time domain. Each accelerometer icon represents a waveform that lasts for about one second and can have g forces of up to ±10 g. These accelerometer data are sent to the UUT computer's multifunction card analog output at a rate of 4 KHz (therefore, 4000 data points are output over a one-second time frame) essentially making the UUT computer into an arbitrary waveform generator. After selecting a specific accelerometer waveform icon, the operator may place it on the accelerometer Time Line by clicking and holding down the left mouse button and dragging the mouse to the location in time that is desired for the accelerometer event. A temporary vertical cursor line is drawn to display the mouse location to allow more precise positioning (relative to the BIT, temperature editor data as well as time) of the accelerometer waveform. Releasing the mouse causes the selected accelerometer icon to be drawn (starting at the location of the mouse), sets the accelerometer event to begin at that time, and erases the temporary vertical cursor line. The vertical size of the icon in the accelerometer editor area represents the peak g force of the waveform. Multiple accelerometer icons of any type may therefore be placed (non-overlapping) in the accelerometer Time Line region.

The BIT Time Line Editor allows the operator to manipulate the time at which a BIT fault report can occur during a scenario lasting up to a minute. Selecting either the "OK" or "Fault" option in the BIT menu causes the BIT Time Line Editor to become active. Clicking and holding down the left mouse button and dragging the mouse to the right draws a horizontal line for the selected BIT report ("OK" or "Fault"). Releasing the left mouse button sets the last position in time for that BIT selection. (In the case of a "Fault" BIT selection, releasing the left mouse button causes the "Fault" BIT report to return to "OK" at the release point. This allows faster and more convenient editor control.) The length of time that a scenario runs is controlled by how far in time the BIT data is located (may be up to a minute in time). Therefore, the BIT Time Line Editor controls how long a scenario runs (from one second to one minute) regardless of the data in either the temperature or accelerometer editors. This allows flexibility in controlling the scenario time instead of being locked into a one-minute maximum for all scenarios.

**3.1.2.5 Run Scenario Exec** – The user may choose to run a scenario by selecting from the Run option in the main menu bar. This opens the selected "run" file, which, in turn, opens the LRU scenario data, environmental, and BIT time line files. The data contained in these files are then used to set up several databases for temperature, accelerometer, and LRU scenario data. The temperature and accelerometer event data are read from their databases and sent to an analog

multifunction card to create the simulated sensor signal outputs. A temperature or accelerometer data event is output only when the time of the event matches the scenario time. The LRU scenario data are sent to the LRU BIT computer (via RS-232C) just prior to the start of the scenario. The information sent to the LRU BIT computer contains the initial, fault, and final values of a typical message that the simulated LRU (in this case, a SCADC) would respond with on the MIL-STD-1553B bus when interrogated by the Smart BIT computer. During the scenario, the UUT computer controls the fault behavior for the simulated LRU by sending RS-232C messages to the LRU BIT computer according to the event information loaded into the BIT time line database. Each BIT event message is sent when the time of the event matches the scenario time (concurrent with the temperature and accelerometer data events). If the UUT computer temperature and accelerometer simulated outputs are not selected (via a user menu window control - see UUT Sensor Control Display, Figure 11), the source of the temperature and/or accelerometer signals are taken from the physical sensors during the scenario. The UUT computer simulated temperature and accelerometer signal outputs may also be summed with the physical sensor to allow more flexible scenario data. The selection of any of the above sensor options is independent of the sensor type. Hence, the temperature could be selected to be a simulation from the UUT computer only, while the physical accelerometer sensor can be chosen.




R91-6080-013

Figure 11. UUT sensor control display.

In addition, the Run Scenario Exec provides a scenario cancel (abort) box that allows the user to stop the scenario before it normally would have been completed. The Run Scenario Exec therefore orchestrates the entire scenario process, from allowing the choice of the scenario to controlling all concurrent data events that comprise a scenario.

**3.1.2.6 Interface Drivers** – The UUT computer Interface Driver software is a mix of 'C' and assembly language driver programs. The purpose of these programs is to provide upper-level UUT computer software access to the RS-232C and multifunction card hardware.

The RS-232C driver software consists of calls to the various built-in input and output Microsoft Windows RS-232C functions. During times when the UUT computer is not running a scenario (referred to as "Idle Mode"), the RS-232C driver software periodically sends a message to the LRU BIT computer to determine if it is time for the UUT computer to perform a self-test. If a system-test command has been executed at the Smart BIT computer, the UUT computer RS-232C self-test request messages to the LRU BIT computer would indicate that a self-test is requested. After performing the self-test, the UUT computer would reply with the self-test results message to the LRU BIT Computer via the RS-232C driver software.

The RS-232C driver software is also used to control the LRU BIT computer before, during, and after running a scenario. Just prior to running a scenario, the UUT computer RS-232C driver software downloads the contents of the LRU scenario generation text file to the LRU BIT computer. This file contains the initial, fault, and final values for the BIT word and any LRU parametric data words. This transmission is responsible for controlling both the type of LRU to be simulated as well as its fault values. During a scenario, the UUT computer "OK" and "Fault" BIT report event data that is read out of the BIT time line database is sent via the RS-232C driver software to control the LRU behavior of the LRU BIT computer. The LRU BIT computer reflects this UUT computer control of the LRU data whenever the Smart BIT computer requests data from the LRU BIT computer over the MIL-STD-1553B bus. Once the scenario has completed, a "Halt" command is sent via the UUT computer RS-232C driver software to the LRU BIT computer to stop the scenario.

The temperature and accelerometer analog signal outputs to the TSMD sensor card (residing in the LRU BIT computer chassis) may be simulated by the UUT computer, or may be generated by the physical temperature and accelerometer sensors, or may be a combination of both

simulated UUT computer output and physical sensors. The sensor signal selection is controlled by the UUT computer-driver software and the UUT computer's multifunction card.

The simulated LRU power-on voltage signal is also controlled by the UUT computer driver software. This signal is used to indicate to the TSMD computer that power has been applied (or shut off) to the simulated LRU (this signal is simulated because power is always applied to the LRU BIT computer). The TSMD computer senses this signal to determine when a scenario has started or been completed.

When a scenario is running, a temperature data word is output to the multifunction card every 1/18th of a second by the UUT computer Interface Drivers software. This provides smooth transitioning of data over time. The output of the accelerometer waveforms is considerably more complex. Each instance that the scenario time matches the accelerometer event start time in the accelerometer database, the UUT computer assembly language driver is called to set up the multifunction card timer to cause interrupts at a 4-KHz rate. The type and length of the accelerometer waveform data are also passed to the driver program. Each time the multifunction timer interrupts the UUT computer, another data word in the accelerometer waveform data table area is sent to the analog output DAC (digital-to-analog converter channel). This continues until all data words have been sent to the multifunction card. (The accelerometer output of the multifunction card is sent to the TSMD sensor card residing in the LRU BIT computer and then sent to the TSMD computer for further processing.) Once all accelerometer waveform data words have been output (total waveform time is about one second), the multifunction card timer is shut off and the interrupts are shut off. The process starts over again whenever the current scenario time matches another accelerometer event start time.

## 3.2 LRU COMPUTER AND SENSOR HARDWARE

The LRU BIT computer provides flexible, realistic rapid-prototyping capabilities of a fielded avionic LRU for effective Smart BIT/TSMD integration and demonstration. The LRU BIT computer is programmed to simulate a Standard Central Air Data Computer (SCADC) which is currently active in the US Air Force inventory and employed on several aircraft. The simulated SCADC BIT responses on the MIL-STD-1553B Bus interface is controlled via RS-232C communications from the UUT computer. The LRU BIT computer also contains the TSMD sensor card, which interfaces with the temperature and accelerometer sensors, and contains the signal conditioning circuitry to provide the following conditioned sensor signals: shock peak,

vibration peak, vibration integral, acceleration, temperature, and voltage. Collectively, the LRU BIT computer performs the following functions:

- Provides LRU BIT Card fault flag simulation
- Simulates LRU MIL-STD-1553B bus interface transfers to report BIT status to the Smart BIT computer
- Communicates with the UUT computer over an RS-232C interface to receive BIT scenario files and control words to coordinate BIT status reports
- Provides TSMD sensors and signal conditioning circuitry
- Provides an interface to the UUT computer to allow sensor signal simulation
- Provides an interface to the TSMD computer to allow the TSMD computer to control and read the conditioned sensor signals.

### 3.2.1 LRU BIT Computer

The LRU BIT subsystem consists of a VME-based microprocessor system with a MIL-STD-1553B bus interface capable of controller, remote terminal, and bus monitor functions. This system provides the capability of emulating any avionic LRU which communicates on the MIL-STD-1553B interface. This approach provides great flexibility for emulating a variety of different fielded LRUs.

The LRU BIT computer contains all of the development tools required to modify the BIT circuitry emulation software. Also, the emulation approach does not constrain the evaluation to a single, peculiar LRU; any LRU employing a MIL-STD-1553B interface could be emulated, providing much greater flexibility than that of a particular avionic LRU. This approach also does not preclude the emulated LRU from being replaced at some future time with actual, fielded MIL-SPEC hardware.

The LRU BIT computer consists of a commercial standard 68020 CPU, floppy/hard disk interface and peripherals, an internal power supply, and a MIL-STD-1553B bus interface capable of remote terminal, controller, and bus monitor functions. This selection also makes the development of the sensor board relatively easy: a wire-wrap card is used for mounting the sensors' signal conditioning circuitry. This also allows easy modifications to the circuitry, which may be useful for assessing and evaluating future modifications and additions to the test bed.

The chassis contains a 3-1/2 in. floppy drive and a 53-Mbyte hard disk, which is the minimum configuration for the selected operating system. This provides adequate space for the development software and LRU emulation software for many different scenarios and LRUs. The floppy disk allows software and/or data to be transferred, and files from the hard disk to be backed up.

**3.2.1.1 BIT Card Emulator** – The Smart BIT/TSMD Integration Statement of Work (SOW) requires an "avionics Line Replaceable Unit, or its equivalent" (Paragraph 4.1.3.1.1 LRU Card Assembly of SOW). The LRU BIT hardware selected for this project is a VME-based microprocessor system which is used to simulate an SCADC LRU.

This selection was made for several reasons. LRUs, necessary spares, support equipment, engineering documentation, and other related items that would be required to fulfill the terms of the Smart BIT/TSMD Integration contract were found to be unavailable or so expensive that they were beyond the cost scope of this contract. Any LRU provided by the government would be GFE and issues surrounding contractor treatment of GFE, such as restoration to original configuration, made this option unattractive. Thus, cost and logistics considerations dictated simulating the LRU with a commercial computer. Additionally, this approach allowed the flexibility to modify the test bed for other applications such as expanding the test capability to include other LRUs. MIL-STD-1750A computers were considered but were found to be quite costly, especially since all would require the purchase of extensive development systems to support Ada or JOVIAL programming languages. (VME MIL-STD-1750A computer boards are available from several manufacturers and could be added to the selected system at a later date.) The system proposed includes its own 'C' development tools ('C' was specified in Paragraph 4.1.3.2 of the SOW as the preferred language for this contract) and the cost of the hardware purchased would meet the cost targets of the contract.

The VME computer system, in conjunction with the OS-9 real-time multi-tasking operating system, provides features to easily develop simulations of other MIL-STD-1553B Bus LRUs:

- The complete software development environment is resident; no additional tools or resources are required to modify the delivered application software. This means that application software can be modified to simulate additional LRUs without requiring additional development resources

- The LRU BIT computer application software, which simulates the SCADC MIL-STD-1553B Bus data suites, is down-loaded and controlled by the UUT computer. The SCADC data suites can be changed by editing the UUT computer time line; no LRU BIT software modifications are required
- The LRU BIT computer application software is written to handle all of the MIL-STD-1553B Bus subaddresses, not just the few used by the SCADC, providing a more generic, flexible simulation
- The LRU BIT computer uses an IEEE-P1014 (VME bus) standard bus structure. U.S. Air Force inventory printed wiring assemblies employing this standard bus that are mechanically compatible can be inserted in the VME chassis.

**3.2.1.2 Sensor Card Assembly** – The sensor card is a VME wire-wrap card containing the sensor signal conditioning circuitry, designed to meet the implementation requirements of this contract. A logic diagram of the TSMD sensor board is shown in Figure 12.

Initially, the accelerometer sensor was required to be capable of being mounted on the board, in all three major axes of the sensor card. However, through the technical reviews, the requirement was changed to locate the sensors remotely from the board to allow them to be stimulated without exposing the entire LRU BIT computer to extreme temperatures and vibration/shock conditions. This resulted in both the temperature and accelerometer sensors being mounted on an aluminum cube, with screw mounting holes drilled and tapped on three sides to easily mount the sensors in three axes.

The requirement for superimposing environmental transducer signals (temperature and acceleration sensors) on top of computer-generated simulations of these transducer signals to the TSMD signal conditioning circuitry was also modified. This requirement was derived from the desire to provide TSMD stimulus synthesized by the UUT computer, direct physical excitation of the sensors, and use a combination of the two to superimpose the UUT computer simulation with physical excitation.

To understand the problem with superimposing the signals, it is necessary to consider the modes of operation of the sensors, which are different. This indicates that the approach used for the temperature sensor should be different from the approach used for the accelerometer. The sensors' operational modes are described as follows:

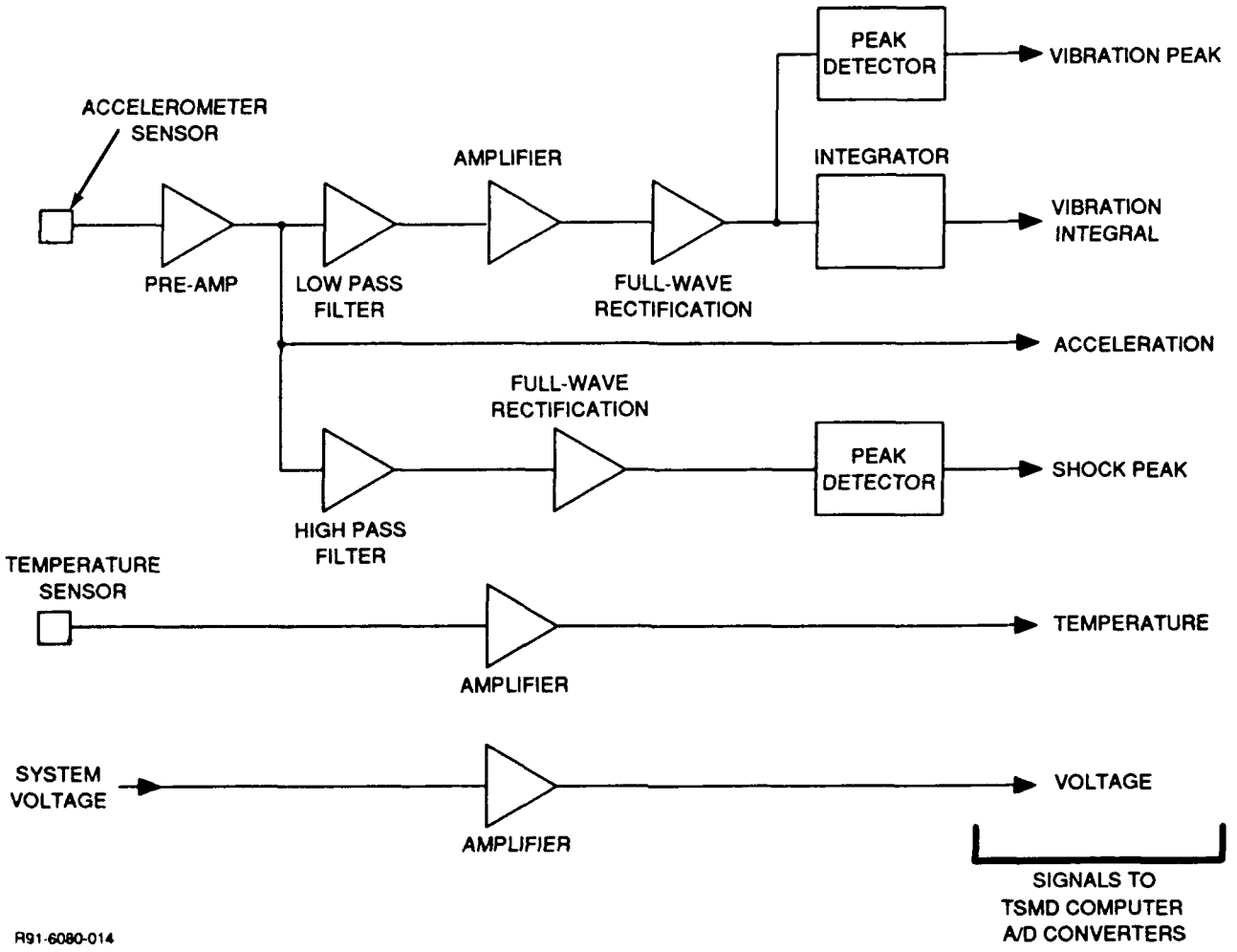



**Figure 12. TSMD sensor board logic diagram.**

- Temperature Sensor. The temperature sensor is a two-terminal, solid-state integrated circuit device which produces an output current proportional to absolute temperature. The current is passed through a resistor to develop a voltage which is sampled by the TSMD A/D converter. Hence, an output is always present which changes in magnitude as the temperature changes
- Accelerometer Sensor. The accelerometer sensor is a self-generating piezoelectric transducer which produces an output proportional to the mechanical energy applied. Hence, in an environment where no stimulus is applied, the output produced is zero; only transient events produce a current.

For evaluation and assessment of the TSMD techniques when using the test bed system, there are three modes of operation for the temperature and accelerometer data:

- UUT computer simulation only
- Physical excitation only
- Physical excitation superimposed on UUT computer simulation.

Figure 13 illustrates the differences between the summer only, mux only, and combined circuit.

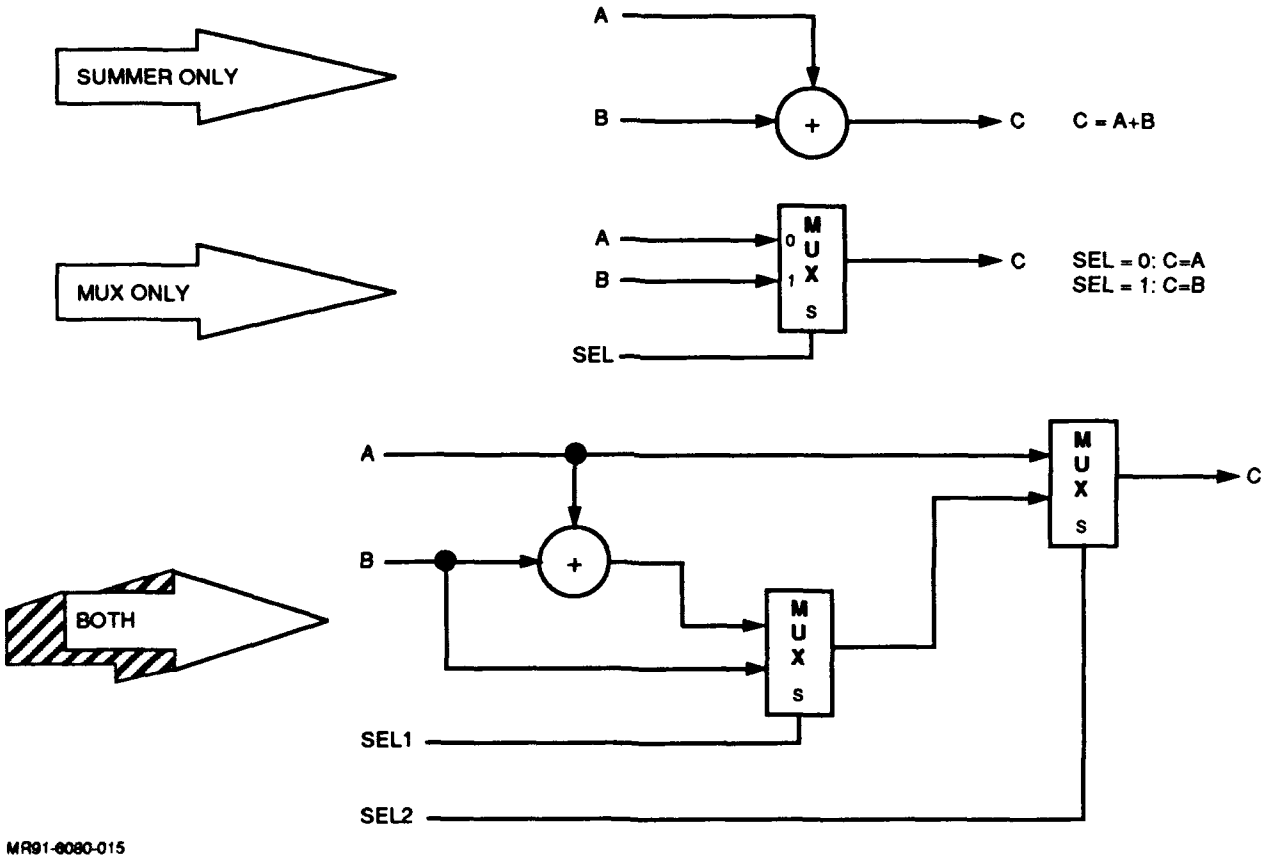



**Figure 13. Signal conditioning control circuitry.**

The UUT computer simulation mode is the primary mode, and is the mode that was demonstrated at the end of the contract. The goal was for the TSMD computer display to directly track the UUT computer simulation timeline, such that TSMD processing can be exercised in a precisely controlled fashion. This mode provides repeatable input data to the TSMD processing, such that controlled experiments can be performed to evaluate and assess the

TSMD processing techniques without having to account for the effects of varying input data. Without a repeatable means of stimulating the signal conditioning circuitry, the results of testing would be inconclusive. This mode of operation cannot be achieved with a 'SUMMER' (superimposed) connection, since the second input could unexpectedly be contributing to the signal during the scenario.

The physical excitation mode is used to evaluate the TSMD techniques with real-world stimuli. Although the UUT computer simulation is very useful for laboratory experiments in assessing and evaluating TSMD techniques, physical stimulus would still be the ultimate method of conclusively exercising the techniques. Simulation cannot completely reproduce the physical sensor signals due to the limited spectral content that the UUT computer analog signal can simulate. Therefore, the physical mode of excitation is highly desirable. This mode of operation also cannot be achieved with a 'SUMMER' connection, since the second input could unexpectedly be contributing to the signal. This potential for the input signal to be corrupted by an anomalous signal on the other input of the summer should be avoided if at all possible.

The third mode of operation is to have physical excitation superimposed on the UUT simulation. This mode of operation provides the capability of having the UUT computer provide a background stimulus (simulated sensor signals), with some physical excitation applied to the sensor being added to the background stimulus. A possible use of this mode would be for the UUT computer to provide background vibration representing aircraft vibration, with shocks being injected via the accelerometer sensor. Though this could also be achieved by modifying the sensor timeline profile, physical excitation allows totally uncorrelated stimuli to be applied to the TSMD processing techniques in a very free-form (and thus a more difficult to control) manner. Although the physical anomalies would provide interesting data to the TSMD, consideration should be given to controlling the experiment and achieving repeatable results by supplying repeatable physical excitation. Consideration should be given to using a controlled and quantifiable method of excitation, such as a temperature chamber and vibration test fixture. Gross manual forms of excitation, such as pencils tapping the accelerometer and ice cubes on the temperature sensor, may be hazardous to the sensors and signal conditioning circuitry and should be exercised with caution.

The temperature sensor provides an output current proportional to absolute temperature. If a 'SUMMER' configuration is used on the signal conditioning circuitry input, then the UUT

computer simulation is applied as an offset, based on the ambient temperature. If a 'MUX' configuration is chosen, either UUT computer simulation only or the physical temperature sensor only (depending on the specific selection) supplies an absolute temperature value.

The accelerometer, which provides a self-generating output, could readily employ either 'SUMMER' or 'MUX' signal conditioning circuitry input connection. The 'SUMMER' connection could be used with the caveat that no physical stimulus be applied when the experiment requires that the UUT computer applied simulation track exactly with the TSMD measurements. Though it may sound simple and easy, it could prove quite impossible to provide an environment free of physical stimuli. Cooling fans, air conditioning systems, airplanes flying nearby, and even people walking by may provide sufficient stimulus to cause an offset to the UUT computer simulation. Using a 'MUX' connection would guarantee repeatable input data when the UUT computer simulation has been selected. The other 'MUX' selection is the output of only the physical accelerometer sensor.

The implementation of these modes was achieved by 'MUX'ing the various input signal paths by using inexpensive 5-volt DIP reed relays, as shown in the sensor board block diagram, Figure 14. The UUT computer controls these relays by asserting the appropriate digital output signals on its multifunction card. The temperature and accelerometer signal data sources may therefore be selected individually by the UUT computer.

The relay control signals were configured such that disconnection of the UUT computer cabling defaults to the physical sensor being connected directly to the TSMD signal conditioning circuitry. This circuit was implemented with single-throw, double-pole 5-volt relays to perform the MUX function. An op-amp adder circuit was used to perform the 'SUMMER' function.

The voltage sensing circuitry is also located on the sensor card, which simplifies cabling between the various test bed hardware elements. This implementation also provides the flexibility of adding specialized voltage conditioning circuitry in the future, which may be desirable for monitoring transients, current surges, sags, and other voltage anomalies which might otherwise go undetected.

The UUT computer uses a digital I/O signal to simulate LRU power ON/OFF state transitions. When in the ON state, the TSMD computer voltage sensing circuitry measures the

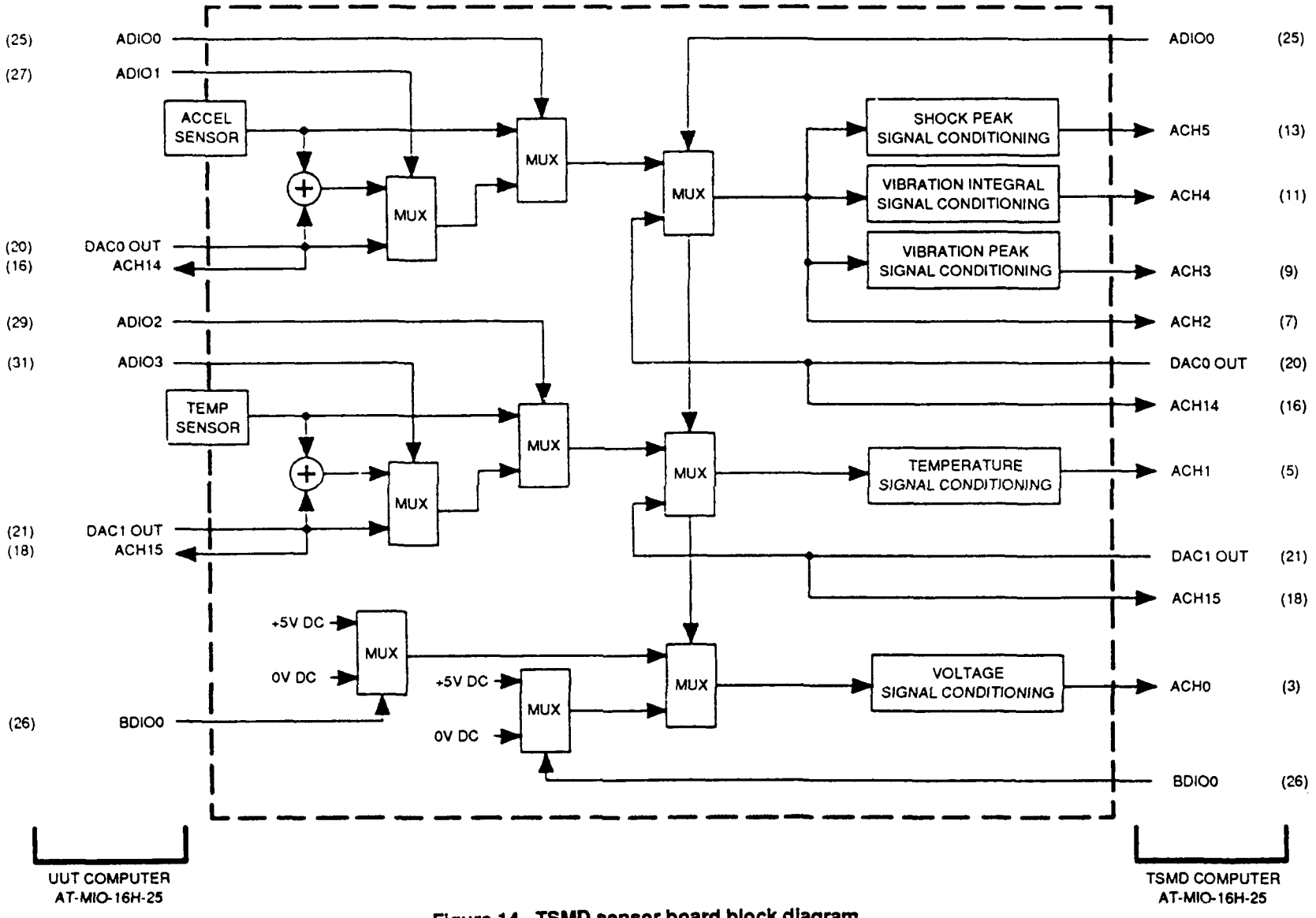



**Figure 14. TSMD sensor board block diagram.**

+5-volt power supply of the LRU BIT computer. When in the OFF state, the TSMD computer measures 0 volts, indicating a power-OFF condition. The TSMD computer uses this signal to initiate sensor signal sample processing and stop processing when a power-OFF state has been detected. This simulates the TSMD activity as it would occur in an actual LRU as power is cycled.

Self-test capability is provided to allow both the UUT computer and the TSMD computer to test the cabling to the sensor board. This is implemented by providing a feedback path for all analog signals such that the stimulus can be measured. The UUT computer is able to monitor both D/A outputs used to simulate acceleration and temperature. The TSMD computer is able to independently stimulate both the acceleration and temperature signals in order to verify operation of the sensor signal conditioning circuitry.

Complete engineering drawings of the TSMD sensor board have been delivered to Rome Laboratory as part of this effort.

*3.2.1.2.1 Temperature Sensor Approach* – Previous sensor developments show two particular characteristics of temperature stress that deserve attention: absolute temperature value and temperature change rate. Both of these characteristics may be monitored readily using a single conditioning circuit, with the change rate data being derived on the TSMD computer using the absolute temperature signal samples.

The selected temperature sensor is Analog Devices' Two-Terminal IC Temperature Transducer, Part Number AD590. Available in several packages, it has the following specifications:

- Linear Current Output: 1 mA/°K
- Wide Range: -55 to +150°C
- Laser Trimmed Accuracy: ±0.5°C
- Linearity over full range: ±0.3°C

Features include:

- The sensors may be remotely located hundreds of feet from receiving circuitry using well-insulated, twisted pair wire
- Costly transmitters, filters, lead wire compensation, and linearization circuits are all unnecessary

- Laser trimming ensures interchangeability
- Excellent power supply drift and ripple rejection
- Electrically durable, such that voltage polarity reversal does not damage the device.

This device has been used previously in both the F-16 TSMD IR&D program and the MicroTSMD contract. As such, it has been used to collect data during actual flight test conditions. The device exceeds the requirement of -40 to +100°C operation.

*3.2.1.2.2 Accelerometer Sensor Approach* – Selection of an accelerometer sensor is not a trivial task. The sensors come in a variety of damping factors that determine the flatness in response at low frequencies (where the vibration characteristics reside) versus the resonant frequency gain at higher frequencies (where shock information is found). Mounting features, sensitivity, frequency ranges, and other design criteria were evaluated.

The accelerometer conditioning circuitry is a much more complex issue than the temperature circuitry. Westinghouse TSMD experience has shown that several characteristics of the sensor must be processed by analog circuitry prior to conversion to digital data if digital signal processing is not employed.

Frequency discrimination is necessary to separate shock and vibration information from the spectral signal output from the accelerometer sensor. This discrimination is readily accomplished by bandpass filters implemented with operational amplifiers and inexpensive discrete components. Based on the analysis of F-16 aircraft operational flight characteristics (detailed in the MicroTSMD final report from Westinghouse, see Appendix G), the F-16 TSMD IR&D spectral filtering used a bandpass filter of 50 to 250 Hz for vibration and a 50 KHz high-pass filter for shock. The bandpass frequency range for the Smart BIT/TSMD integration contract is 5 to 5000 Hz.

In addition to the discrimination of vibration, the F-16 TSMD research determined that both peak and integral values were of sufficient interest to be measured over a periodic time interval. The one-second interval used in the F-16 TSMD flight test provided a useful starting point. This time interval is controlled by the TSMD computer, allowing the time interval to be flexibly programmed via software. The peak detection circuitry uses captured transients that would normally go undetected by means of sampling A/D values using the TSMD computer performing

digital signal processing algorithms. Likewise, the integral of the vibration discriminated signal provides vibration density information that a sampling approach could not achieve. To meet the above stated accelerometer requirements, an Endevco Model 2221D sensor was selected. It provides a frequency response of 2.0 to 8.0 KHz (linear region) and a thru-hole for easy mounting.

All of these capabilities are provided by the TSMD sensor board signal conditioning circuitry. The schematic for this board has been provided to Rome Laboratory as part of this effort.

### 3.2.2 Test Vehicle (LRU) Software

The LRU BIT test vehicle software is written in the 'C' programming language and runs under the OS/9 operating system. The program's primary function is to simulate the target LRU's MIL-STD-1553B bus traffic as directed by the UUT computer. A function diagram is shown in Figure 15. The software is designed to be easy to maintain and/or modify to simulate LRUs other than the SCADC. In addition, the LRU BIT computer contains its own development system, with tools included to maintain and modify the simulation program. These capabilities are available using the TSMD computer as a terminal.

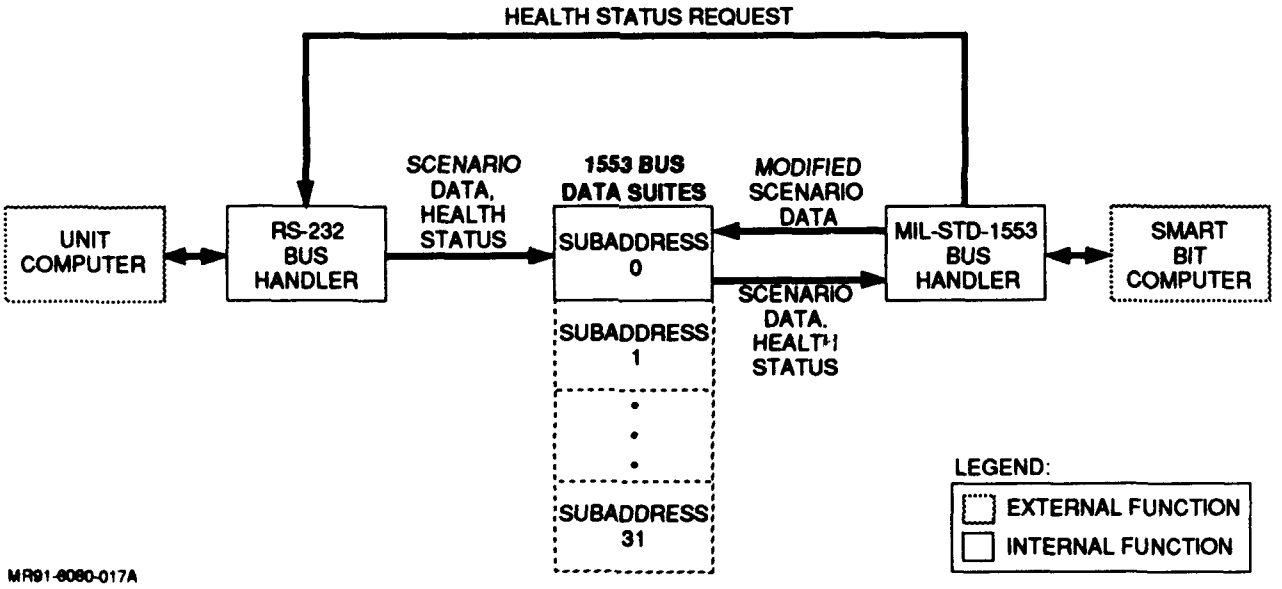



**Figure 15. LRU BIT computer functional block diagram.**

When power is first applied to the LRU BIT computer, the test vehicle software performs system self-tests and initializes its operating environment from the batch files, programs, and data files located on its hard disk. Once properly powered up, it loads the main program. The LRU BIT computer block diagram is shown in Figure 16. This program consists of two modules: M1553_HAND, a 1553B bus management facility, and RS232_HAND, an RS-232C command interface facility.

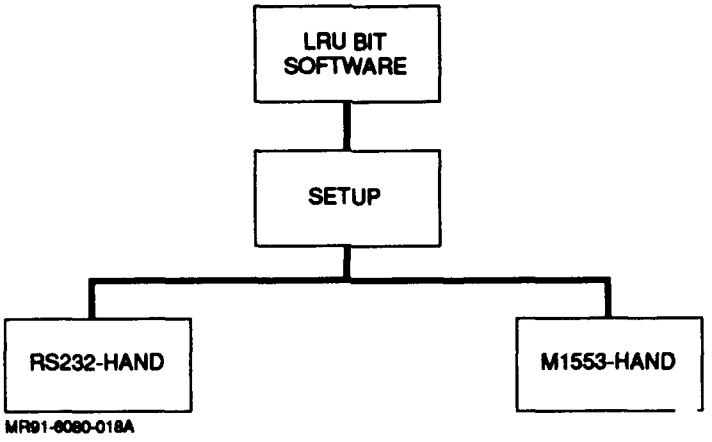



**Figure 16. LRU BIT computer software block diagram.**

The LRU BIT computer chassis contains a COTS MIL-STD-1553B bus driver card. This card contains high-level software driver functions in firmware, eliminating the need to write low-level drivers. This card therefore provides a well-behaved platform for communicating with the Smart BIT computer when simulating the LRU. The test vehicle software simulates a MIL-STD-1553B remote terminal (the SCADC) by responding to MIL-STD-1553B queries for data using "canned" data that have been downloaded (prior to running the scenario) from the UUT computer. The test vehicle software must also communicate with the UUT computer via the RS-232C interface while MIL-STD-1553B messages are being exchanged with the Smart BIT computer. This is achieved by making use of the stable COTS RS-232C hardware and software supplied by the manufacturer (Radstone).

The test vehicle software's primary function is to simulate the MIL-STD-1553B bus traffic of a SCADC or other LRU with the data supplied by the UUT computer. To achieve this, the software interprets the RS-232C commands received from the UUT computer and supplies the MIL-STD-1553B bus card with LRU data. When a scenario is running, this information is

transmitted to the Smart BIT computer upon receipt of a MIL-STD-1553B bus transmit command. The test vehicle software is organized into two functional modules, the RS-232C handler and the MIL-STD-1553B handler (see Figure 15). The RS-232C commands are divided into on-line and off-line functions. The off-line functions include setting up test scenario data suites with initial data values, fault data values, incremental data values, and maximum values for each element of the LRU data suites. In addition, since the UUT computer is not linked directly with the Smart BIT computer, the test vehicle software must signal the UUT computer when a system-wide self-test is being executed. It also receives the UUT computer self-test results and reports this status to the Smart BIT computer. The on-line functions include simulating LRU BIT failures, resetting the scenario, and instructing the test vehicle when to run or halt the scenario. When a scenario runs, the test vehicle software's MIL-STD-1553B bus handler responds to the transmit commands by retrieving data from the LRU data tables (downloaded just prior to the scenario run by the UUT computer) and transmits it to the Smart BIT computer. Any LRU data values that have been marked to change in value are incremented by the specified amount until the maximum value has been reached. This feature is provided to simulate real-world dynamic conditions (i.e., simulate a "flying" LRU).

## 3.3 TSMD COMPUTER

The TSMD computer provides the capability of performing, assessing, and evaluating the TSMD processing algorithms. These algorithms include life-stress monitoring, over-stress event logging, and Fault event logging. The TSMD computer provides the following capabilities:

- Provides an interface with the sensor board in the LRU BIT computer
- Performs the TSMD sensor data collection, processing compaction, and storage functions
- Displays TSMD data to the operator in an easy to understand and flexible format
- Interfaces with the Smart BIT computer via an IEEE-488 parallel digital data bus
- Allows the investigation of the correlation of recorded stress data with LRU BIT failure information
- Provides the ability to assess TSMD effectiveness and capacities.

The TSMD computer is a standard Zenith Data Systems Z-248, augmented with a COTS multifunction I/O card and an IEEE-488 interface card (both from National Instruments). The multifunction card is provided to digitize the analog signals of the sensor board. The IEEE-488 bus interface card provides communications with the Smart BIT computer. The multifunction I/O card supports the inputs generated by the sensor card, and spare inputs are used for

wraparound self-test. The IEEE-488 bus interface card provides a mature and well-behaved parallel communications path for the TSMD and Smart BIT computers.

The software for the TSMD computer is written in the 'C' language and operates under the Microsoft Windows environment. The Windows operating system is the same as for the UUT computer so that the user learning curve for the Smart BIT/TSMD Integration System is somewhat shortened. The TSMD software controls the two communications interfaces (analog sensor data and IEEE-488 digital bus) to provide the required real-time link between the TSMD sensor card and the Smart BIT computer. It also includes several TSMD tools which are useful for evaluating the sensor data received during or after a demonstration scenario.

### 3.3.1 TSMD Computer Hardware

The TSMD computer is a standard 12-MHz Zenith Z-248 computer containing 1.1 Mbytes of memory, hard disk, floppy disk, VGA display adapter and color monitor, Multifunction I/O card, and IEEE-488 bus interface card. Microsoft Windows running under MS-DOS provides a multi-tasking environment and graphical user interface. The TSMD computer software was written in the 'C' programming language using Microsoft's Windows development tool kit. The TSMD processing algorithms were previously developed under the F-16 TSMD board and Micro-TSMD projects. The TSMD user interface function is tailored to respond to commands and queries by the Smart BIT computer over the IEEE-488 bus interface. The display software was developed to provide the operator with the capability to visually assess the performance of the TSMD data collection, processing algorithms, and sensor data storage using Microsoft Windows.

The TSMD computer interfaces with the sensor card (resident in the LRU BIT computer chassis) to receive conditioned analog sensor data. Self-test wraparound capability is provided to assure proper multifunction board communication with the sensor card. An 8-bit IEEE-488 parallel interface is used to communicate with the Smart BIT computer; it consists of COTS hardware and adheres to the IEEE-488 standard. This provides a mature, stable interface for the Smart BIT and TSMD computers to communicate over.

### 3.3.2 TSMD Computer Software

The TSMD computer software is written in the 'C' language and operates under the Microsoft Windows operating environment. The software is written to facilitate the manipulation of TSMD characteristics such as short-term queue length and event buffer size. It is divided

into two programs that correspond to the on-line and off-line functions of the TSMD (see Figure 17). The on-line function monitors, correlates, and records the information received from the TSMD sensor board located in the LRU BIT computer chassis and provides this information to the operator and the Smart BIT computer in real-time. In addition, this on-line function provides the ability to store its accumulated data to the hard disk of the TSMD computer to allow off-line functions to display and generate operator-readable reports from the data stored to disk.

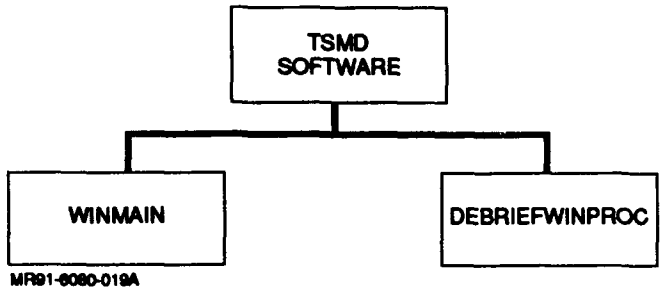



**Figure 17. TSMD computer software block diagram.**

After power is applied, the TSMD computer performs a system self-test. It then initializes the Windows environment using batch files, programs, and data files resident on the hard drive of the system. Once the Windows environment has been initialized, the Main Wnd Proc starts the Clock Wnd Proc and sets up the interrupt structure to respond to two interrupts. See Figure 18 for an overall block diagram of the TSMD computer software. See Figure 19 for a hierarchical software block diagram.

The first interrupt (Real_Time) responds to real-time interrupts generated by the multifunction I/O card. This routine performs data collection and TSMD processing. It monitors the analog TSMD inputs, maintains short-term data arrays, maintains life-stress histographs, logs events, and takes fault signatures when needed. The short-term data queue is a buffer of the latest environmental measurements. The data are read from the A/D inputs and placed in this buffer without modification. Next, each of the readings is compared to the parameter thresholds held in memory to determine the associated bin number. The life-stress histograph is then updated by incrementing the appropriate bin counter. The bin number is also compared to that of the previous readings to determine if it has changed. If it has, the change is noted in the event buffer. Finally, if an LRU BIT failure has been relayed to the TSMD computer (by the Smart BIT computer), the TSMD software waits until the failure time is located in the center of the short-term data queue and then records the short-term data queue as part of a fault signature. The

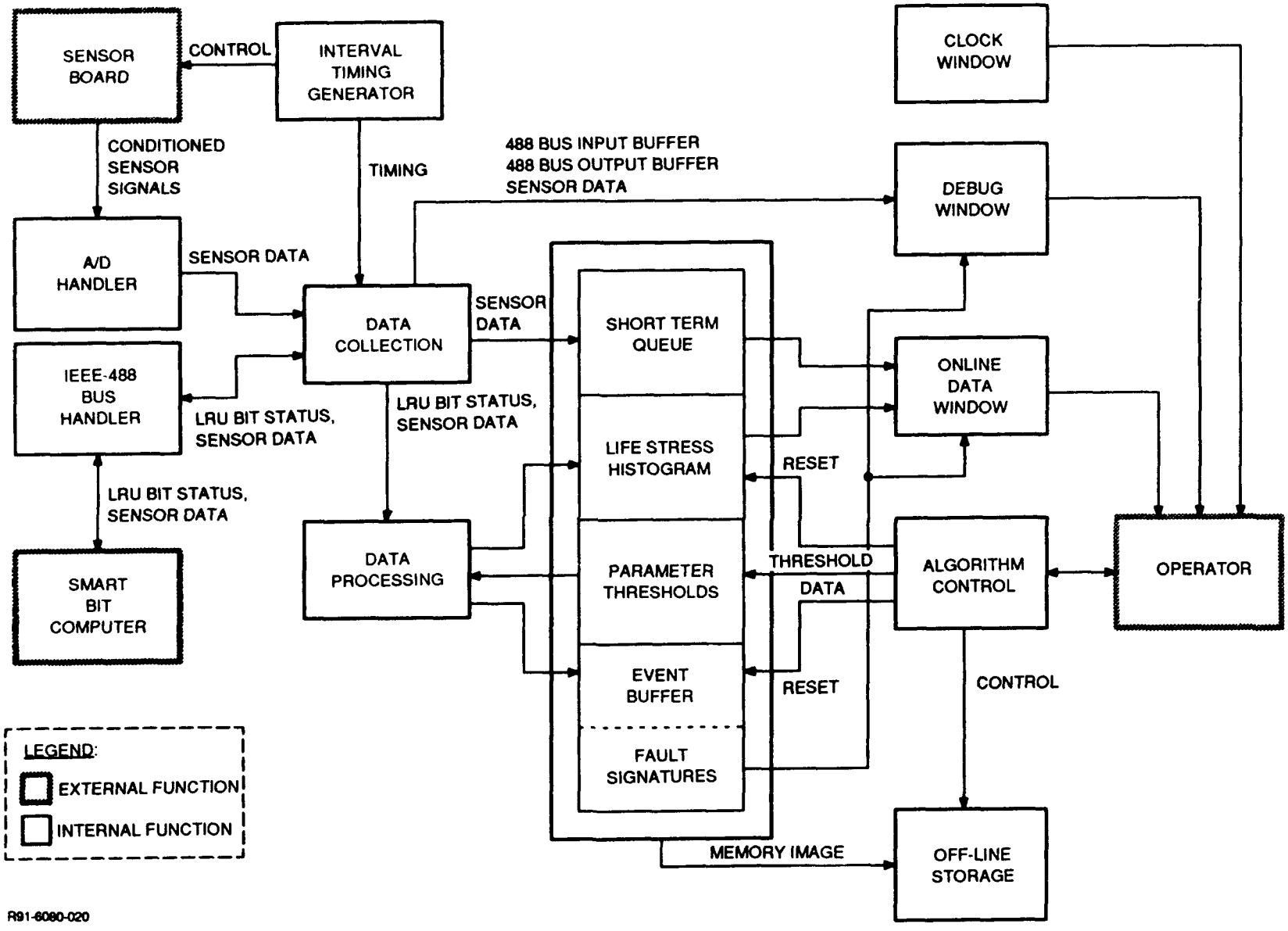



**Figure 18. TSMD computer functional block diagram.**

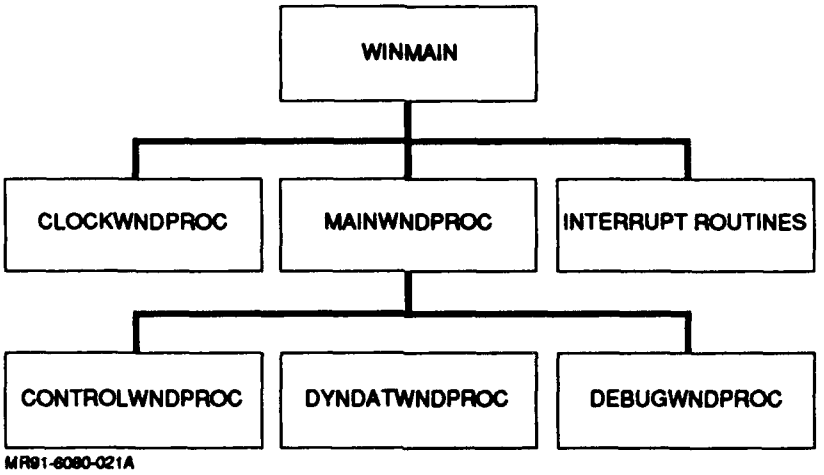



**Figure 19. TSMD display tasks block diagram.**

second interrupt, GPIB_Task, is responsible for handling the IEEE-488 parallel bus which links this computer with the Smart BIT computer. This task is responsible for examining the TSMD data tables set up by the preceding task and providing information to the Smart BIT computer as requested.

The TSMD computer displays the main window, which has menu selections that allow the operator to open Control, On-line Data, and Debug windows as well as manipulate the Clock window at the lower right of the screen. This is illustrated in Figure 20 (with the TSMD Control Panel selected). The Threshold Control window, shown in Figure 21, allows the operator to manipulate the TSMD threshold values for the parameters and thereby control the function of the TSMD sensor simulation. The Memory Image File Control window is shown in Figure 22. It provides the operator with the ability to add logistics information to the memory image prior to saving it to the hard disk. The On-line Data window, shown in Figures 23 and 24, displays real-time data collected in the various TSMD Data structures (both graphical and tabular formats are available). The Debug window, illustrated in Figure 25, shows the analog sensor readings, event buffer, and IEEE-488 communications buffers without any conversion of data and can be used for debugging the hardware and software. These windows operate as independent tasks and allow access to system control and display functions. All of the control and display tasks run concurrently with the multifunction and IEEE-488 interrupts and share data with these tasks within the Windows environment.

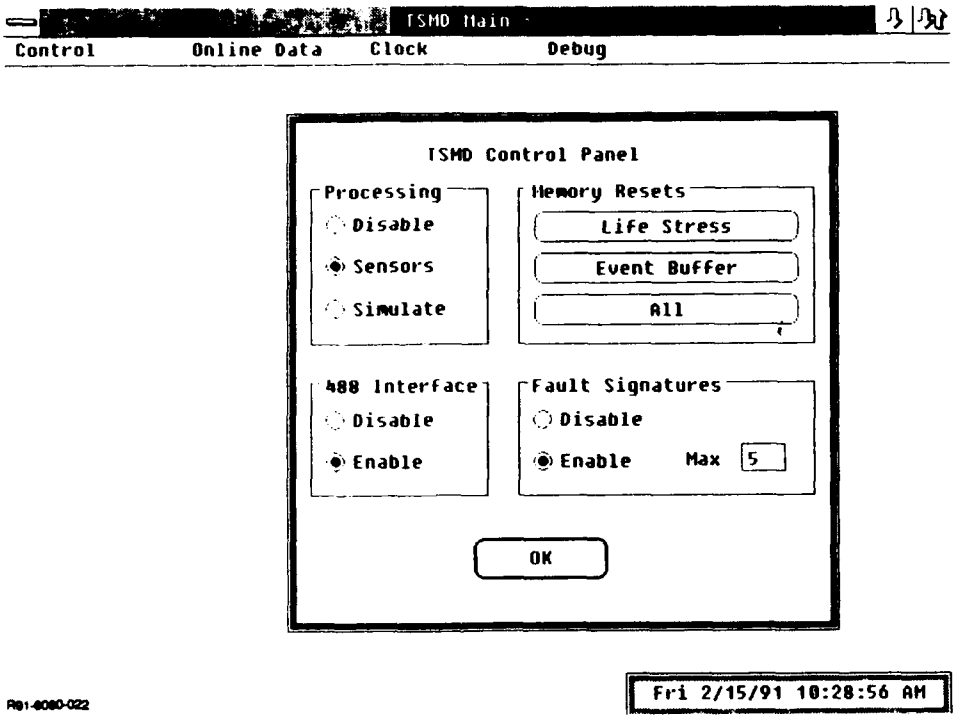



**Figure 20. TSMD computer main window.**

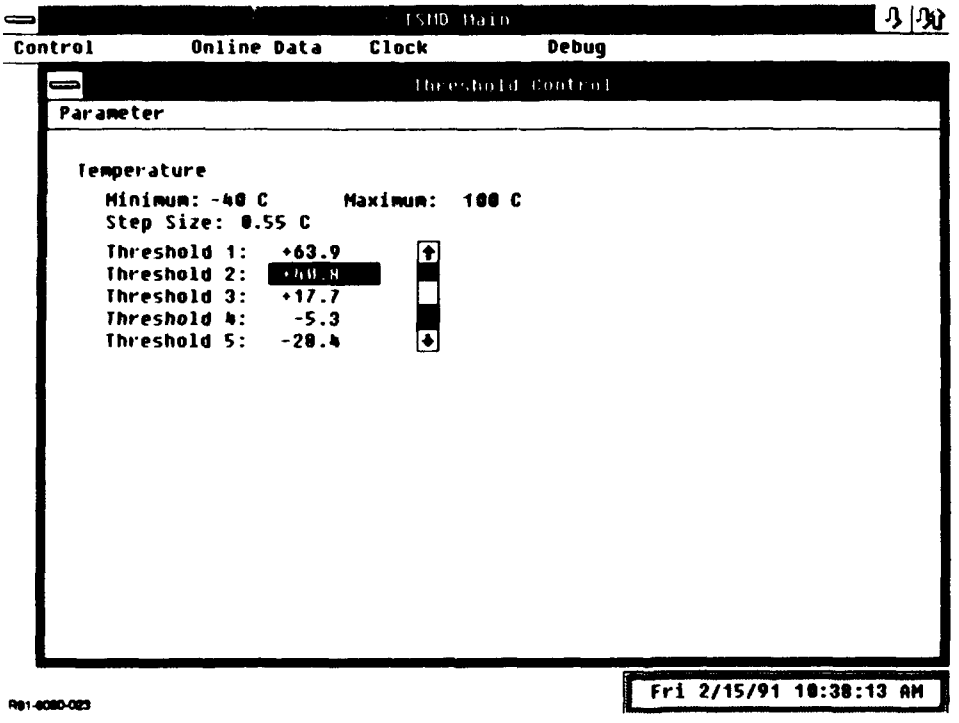



**Figure 21. TSMD computer threshold control window.**

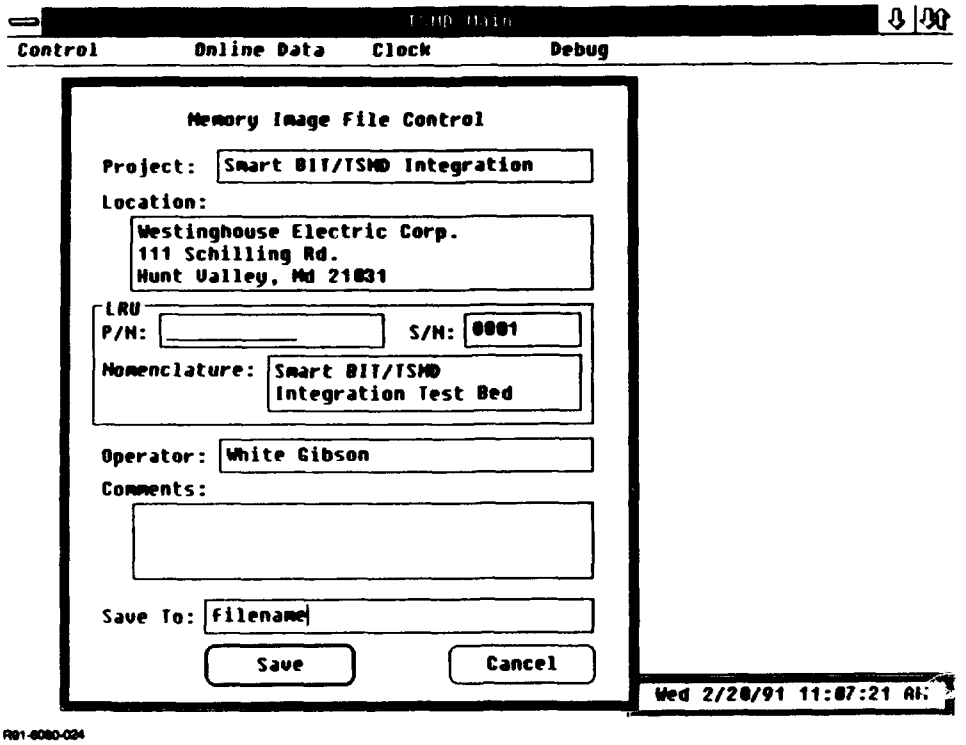



**Figure 22. TSMD computer memory image file control window.**

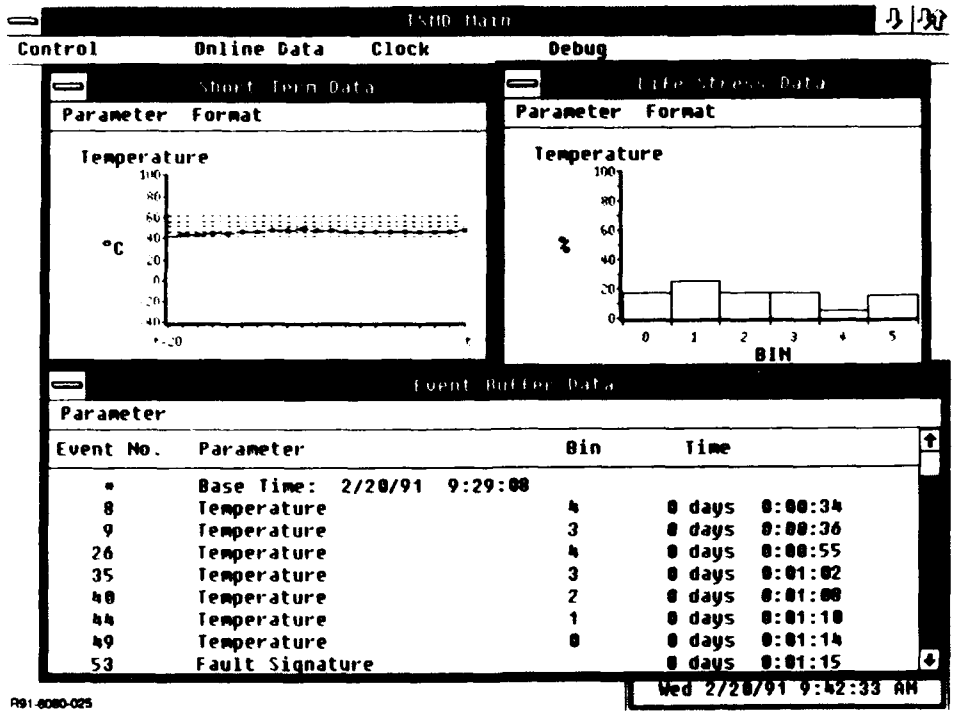



**Figure 23. TSMD computer on-line data window, sample 1.**

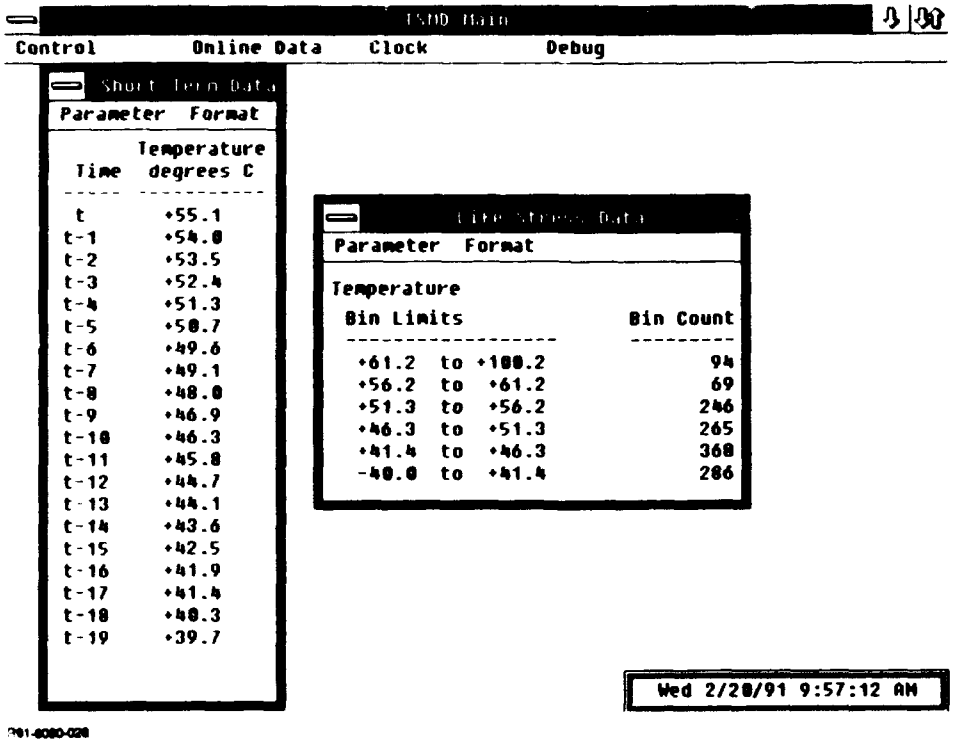



**Figure 24. TSMD computer on-line data window, sample 2.**

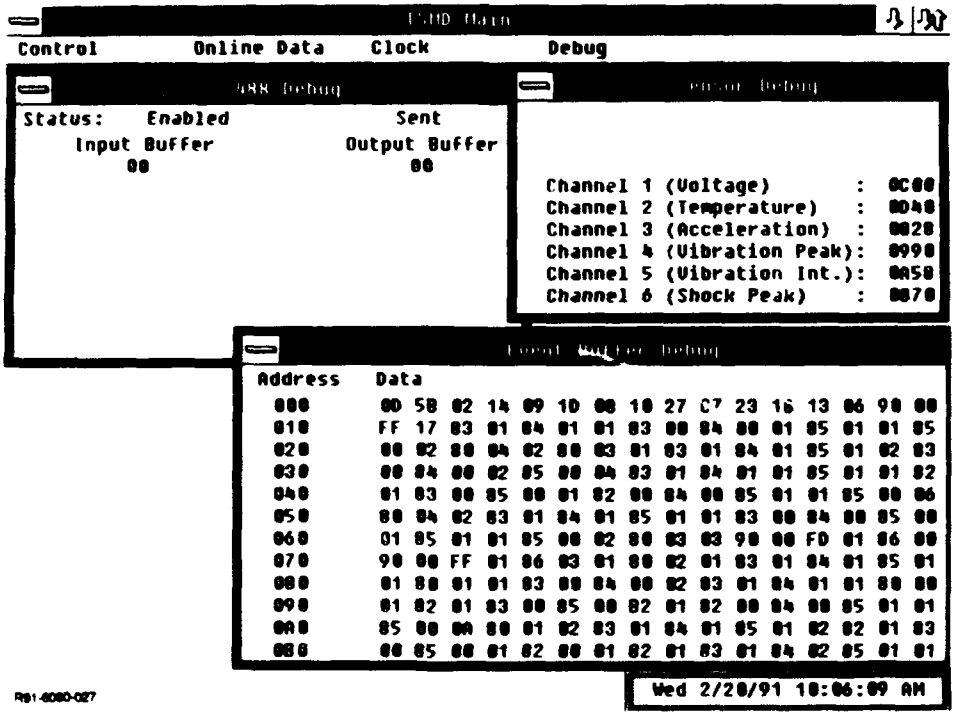



**Figure 25. TSMD computer debug window.**

## 3.4 SMART BIT COMPUTER

The Smart BIT computer is a 25-MHz Compaq 80386 computer. Its function is to analyze the TSMD and LRU BIT data using the Information Enhanced BIT, Improved Decision BIT, Adaptive BIT, and Temporal Monitoring BIT techniques to determine intermittents, false alarms, and hard fault LRU conditions. It graphically displays the Smart BIT techniques while in operation, as well as the LRU BIT data, and displays the results of the Smart BIT processes and their assessment of the UUT health on the Smart BIT display. The LRU BIT and TSMD sensor data may be saved to a disk file for re-running at a later time. A display (when manually activated) of the average Neural Network (Adaptive BIT) error is also provided to show the Neural Network learning as scenario data are processed.

Emphasis has been placed on easy understanding of the Smart BIT display. This was accomplished by color-coding similar function windows as well as the addition of a special status indicator-type window. The BIT status window displays the following labels with a status indicator for each: "LRU BIT Report Status" (LRU BIT report), an "OFP BIT Status" (an assessment of the LRU BIT based on how a typical modern aircraft Mission Computer OFP would react to an LRU BIT fault report), and the "Smart BIT Status" (Smart BIT assessment of the LRU BIT). These indicators (and their labels) are object-oriented and belong to the "Indicator Window" flavor object. They are updated each time an LRU BIT and TSMD message is read in and evaluated by the Smart BIT AI software. This helps to facilitate comparison of the present BIT assessment approach used on a typical modern aircraft with the Smart BIT assessment.

### 3.4.1 Smart BIT Computer Hardware

The Smart BIT computer hardware consists of a 25-MHz Compaq 80386 computer with a 110-Mbyte hard disk drive, 5.25- and 3.5-inch floppy drives, 13 Mbytes of RAM, a VGA-plus color graphics card/monitor with 800 x 600 resolution (with 16 colors), and a three-button high-precision mouse. The Smart BIT computer contains an IEEE-488 bus interface card to provide a parallel data channel with the TSMD computer. The Smart BIT computer also contains a MIL-STD-1553B interface card required for communicating with the LRU BIT computer for monitoring the LRU BIT status. All of the Smart BIT computer hardware are COTS equipment (Commercial-Off-The-Shelf) and, therefore, no hardware design work was required for this subsystem.

### 3.4.2 Smart BIT Software

The Smart BIT software is written in both LISP and 'C' ('C' is used for the interface driver code, only) and is built on the X11 Windows and Unix System V version 3.2 operating system. The Smart BIT software design consists of a main program (see Figure 26), input processes, Neural Networks, K-Nearest Neighbor, Temporal Monitor, and window display programs.

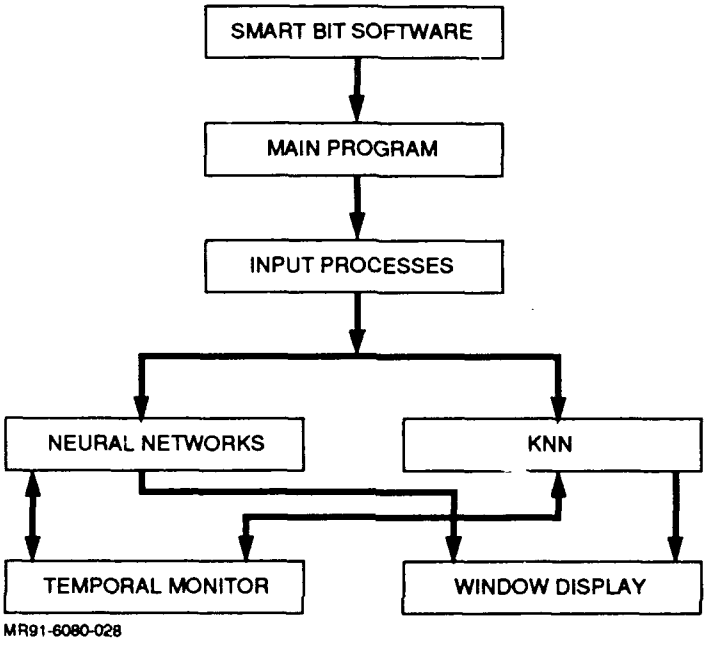



**Figure 26. Smart BIT AI software top-level flow diagram.**

**3.4.2.1 Main Program** – The Smart BIT main program software initially calls functions to load all of the Smart BIT code into the LISP world, and make the BIT Status and Smart BIT Parameter menu windows. It also provides the scenario control code for setting up the source of the data streams (from external hardware or from a disk file) as well as creating the code for the selected Adaptive BIT technique.

Loading the Smart BIT software is a complex task. This is because so much of what is required to set up the Smart BIT software startup is automatically performed by the Unix System V 'shell' programs and LISP initialization files. When the user logs onto the Smart BIT computer, a number of Unix, X11 window, and LISP shell programs are loaded in sequence. After all startup shell programs have been executed, the basic LISP world without the Smart BIT software is loaded into memory. This LISP world contains the flavors, editor, Common LISP,

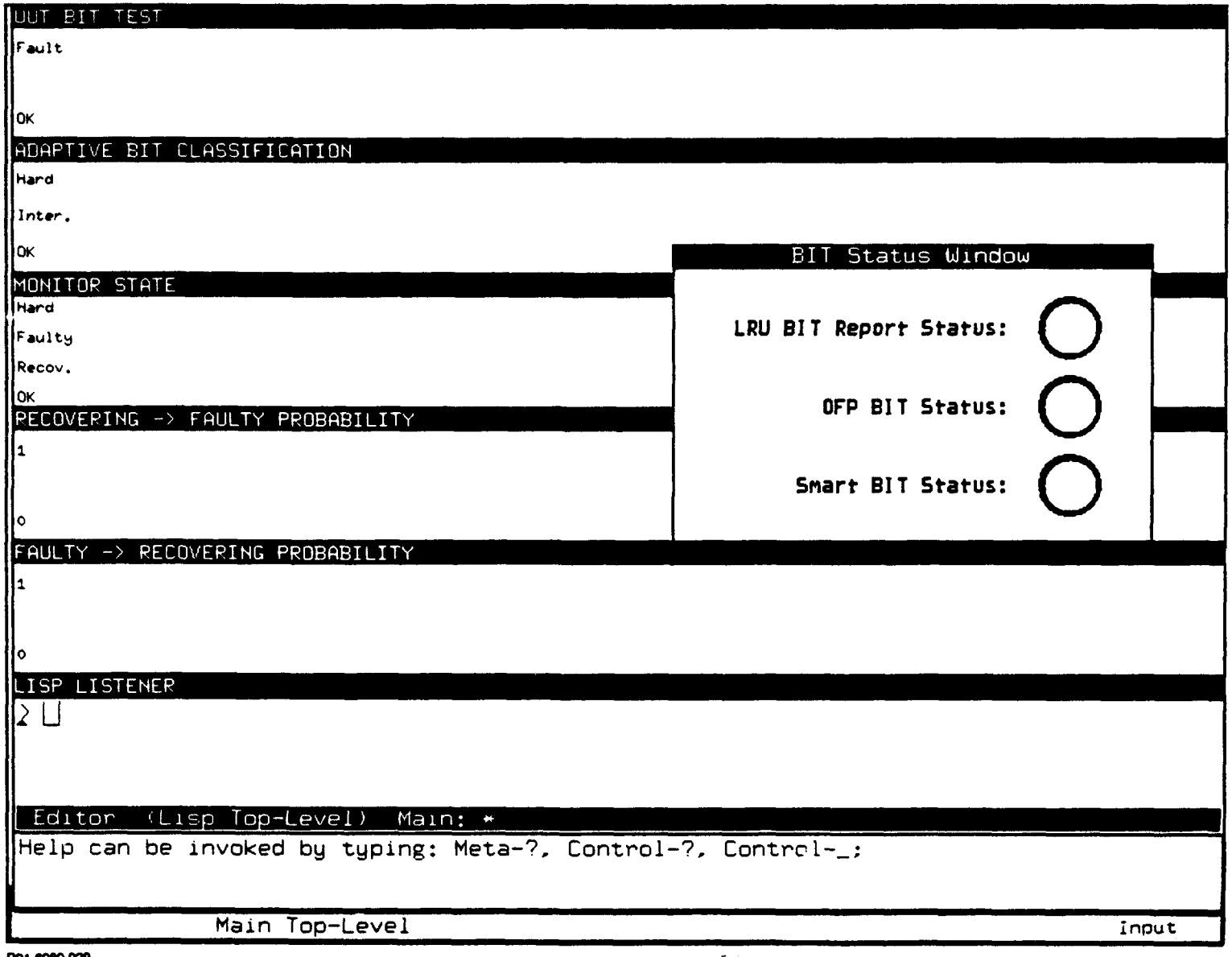
and LISP windows toolkit packages. Finally, the Smart BIT LISP and 'C' interface software are loaded into the basic LISP world. A startup function is called (MAKE-SMART-BIT-SCREEN) to initialize the LISP window system and paint the Smart BIT windows in color onto the Smart BIT computer display.

The MAKE-SMART-BIT-SCREEN startup function initializes the LISP window environment, loads the color registry and text fonts, loads the 'C' 'foreign' function files, sets up the Smart BIT parameters menu window (this menu is a mouse-sensitive parameter window that allows changes to be made to data that affect the operation of the Smart BIT software), creates six line graph windows, creates a BIT status window (the BIT status window displays the LRU BIT, OFP interpretation of the LRU BIT, and the Smart BIT instantaneous status as lamp indicators similar to the kind that a pilot might see), and one LISP listener/editor window for display and keyboard command entry. After all windows have been created, the LISP listener/ editor window displays a keyboard cursor prompt to tell the user that the Smart BIT software is ready for a command to run a scenario (see Figure 27 for the initial Smart BIT display).

When the user runs a scenario, the command line sets up a call to the do-scenario function from the following example command line:

(do-scenario "" :control :bus :learn-by :neural-net)

The do-scenario function uses the ':control' keyword to determine which direction the data are to come from. The ':control' keyword may contain the ':bus' option, which indicates that the source of the data is from the MIL-STD-1553B and IEEE-488 busses. The ':write-file' option keyword performs a similar function to the ':bus' keyword (reading in data from the MIL-STD-1553B and IEEE-488 busses), with the additional feature of data-logging the data from both busses to a specified file (whose pathname would be specified between the "" field in the command line). The ':read-file' keyword option causes the data source to be read from a specified disk file whose pathname would be located between the "" field in the command line. The do-scenario function also initializes the software bus drivers, and then reads in the bus data and saves it to an array. The array bus data are rescaled, collated into a LISP list object, and stored in the message queue. The message queue is then unloaded, with the data processed by the temporal monitor and the Adaptive BIT classification technique that was specified in the ':learn-by' keyword (either neural network or K-Nearest Neighbor) of the do-scenario command line.




R91-6080-029

**Figure 27. Initial Smart BIT display**

**3.4.2.2 Input Processes** – The Smart BIT computer must communicate with both the TSMD and the LRU BIT computers via digital interfaces. The Smart BIT computer requests data from the TSMD computer via the IEEE-488 parallel bus. The Smart BIT computer also requests data from the LRU BIT computer via the MIL-STD-1553B bus. The Smart BIT computer LISP software performs this communication task by calling 'C' driver programs. The 'C' drivers are needed because LISP is ill-equipped to handle the highly structured Unix System V kernel driver protocol that must be followed when software is required to communicate with hardware. Therefore, the Smart BIT LISP software makes 'C' 'foreign' function calls to obtain the IEEE-488 and MIL-STD-1553B bus message data. All of the bus data is placed in a 'foreign array' (byte sized). The Smart BIT LISP software then reads the data and rescales, reformats, and places them on a message queue for further processing.

**3.4.2.3 Neural Network Adaptive BIT** – The Neural Networks software program (when :neural-net option is used by the operator in the do-scenario command line) initially sets up a new instance of the Neural Network and then creates the input, middle, and output neuron layers. The intralayer and interlayer interconnection of the neurons are then set up. Neural Network learning of the BIT and TSMD data then begins by processing the message queue. The message queue data are propagated through the Neural Network to the output layer. An error is then calculated and propagated backwards through the network. This is done several times to reduce the error factor for each input data list. The results of the average error are displayed on the Neural Network average error line graph window (if it has been enabled for display). In addition, the output of the Neural Network is displayed on the Adaptive BIT line graph window. Both the Neural Network average error and the Neural Network output windows are the same color (cyan) to more easily identify these displays as being part of the same Adaptive BIT technique. The Neural Network then passes the same message queue data to the Temporal Monitor so that it may classify the LRU BIT behavior according to a temporal approach.

**3.4.2.4 K-Nearest Neighbor** – The K-Nearest Neighbor program calculates the nearest neighbor algorithm and displays it in the Adaptive BIT line graph window (when the K-Nearest Neighbor algorithm is chosen by the user using the ':knn' option in the do-scenario command line). An instance of the K-Nearest Neighbor object is first created and initialized before any data are processed. After all bus input data (from the LRU BIT and TSMD computers) have been massaged and pushed onto the message queue, the K-Nearest Neighbor object methods determine which cluster group each new unknown input scenario data sample belongs. To

accomplish this, it must determine where its nearest neighbors are by looking at a list of previous samples in the area around the present unknown input scenario data sample. The present unknown input scenario data sample is then assigned the same classification of the nearest cluster of previous samples. This classification is then displayed on the K-Nearest Neighbor adaptive BIT line graph window for each input scenario sample processed. The K-Nearest Neighbor algorithm then passes the same message queue data to the Temporal Monitor so that it may classify the LRU BIT behavior according to a temporal approach.

**3.4.2.5 Temporal Monitor** – The Temporal Monitor software performs its function by first creating an object which contains state and probability data within it. The object itself is created once upon the initial loading of the Smart BIT LISP software. Temporal monitoring is always used in conjunction with either K-Nearest Neighbor or Neural Network (depending on the user's selection). It is therefore called from either the K-Nearest Neighbor or Neural Network code. The temporal monitoring Markov Model Finite State Machine consists of four states: OK, Recovering, Intermittent, and Hard. Transitioning from one state to another is accomplished by calculating the probability of a Fault or OK condition based on past input scenario data.

If the present state is OK and an LRU BIT fault report occurs, the Temporal Monitor changes the state to Intermittent (Faulty). If no LRU BIT fault reports occur, the state remains unchanged (OK).

If the present state is OK, and an LRU BIT fault report occurs, the Temporal Monitor changes the state to Intermittent (Faulty). If no LRU BIT fault reports occur, the state remains unchanged (i.e., in the OK state).

If the present state is Recovering, the Temporal Monitor checks the size of the input scenario sample group that is OK. If the number of consecutive OK samples is greater than a dynamically estimated threshold, then the state is changed to OK; otherwise the state remains in Recovering. However, if a faulty sample is observed, then the state is changed to Intermittent (Faulty).

If the present state is Intermittent (Faulty), the Temporal Monitor checks the size of the input scenario sample group that is Faulty. If the number of consecutive Faulty samples is greater than a dynamically estimated threshold, then the state is changed to Hard; otherwisethe state remains Intermittent (Faulty). If not, and an OK sample is observed, the state is changed to Recovering.

If the present state is Hard, the Temporal Monitor remains in the Hard state. However, if an OK BIT report sample is observed, then the state changes to Recovering.

All four Temporal Monitor states – OK, Recovering, Intermittent (Faulty), and Hard – are displayed in the Temporal Monitor output window. In addition, the transition probabilities for both the Recovering to Faulty and Faulty to Recovering are displayed in two separate windows. All three windows are displayed in the same color (blue) to more easily identify these displays as being part of the Temporal Monitor process.

**3.4.2.6 Window Display** – The window display software contains three window flavor objects. the LINE-GRAPH WINDOW, the MENU-WINDOW, and the INDICATOR-WINDOW. The LINE-GRAPH-MIXIN flavor object is used to build all of the line graph window displays for the BIT, Adaptive BIT, Temporal Monitor, and transition probability displays. Color is a line graph window object attribute inherited by each window to allow line graph windows to be color-keyed to their function, thus providing a more easily understandable user interface.

The LINE-GRAPH methods (object functions) have been specifically setup to draw character strings as well as line graphs (similar to strip charts) in a selected window. When the startup function MAKE-SMART-BIT-SCREEN is called, it makes each of the line graph windows (Adaptive BIT, Temporal Monitor output, etc) by making an instance of the LINE-GRAPH flavor. Each new instance of the line graph flavor window object is initialized by first creating and clearing a bitmap of the chosen size and color. Next, the type of fonts required is set up, its vertical text limits are set, the window title is displayed, and the distance the data may change both in the vertical and horizontal directions (according to the height and width of the window) is calculated. The line graph window objects are then ready for calls to display line graph data for the BIT data, Adaptive BIT, or Temporal Monitor techniques.

The MENU-WINDOW flavor object contains Lucid Common LISP window function (from the Lucid Window Toolkit) calls to produce the Smart BIT parameters menu window that is created upon initial loading of the Smart BIT LISP software. An instance of MENU-WINDOW is created to provide user selection of adaptive BIT parameters such as gain factors or number of neurons for the Neural Network, etc). The menu parameters are mouse-sensitive so that the user may click and choose the selected data field and place new data in its place. In addition, the

MENU-WINDOW flavor object provides the ability to syntax-check the user input for a selected data field. Whenever data are entered after selecting it (by using the left mouse button) to change a value, a specific parameter syntax function is called to check the data entry. If the entry is not valid for that parameter, the terminal bell is made audible, the data field is erased, and the menu cursor is placed at the beginning of the selected field to prompt the user to enter new and valid parameter data. Once valid data are entered (by a carriage return, i.e , "Enter" key), the data are saved into the appropriate variable for use by the Smart BIT software when a scenario is run. Boolean selections such as YES or NO are also used for certain fields. This is useful for selecting whether a window is to be displayed or not when a scenario is run. The selection is shown in bolded print font type when that item has been chosen. The Boolean value of 0 or 1 (depending on which item has been chosen) is then saved in the appropriate variable for use by the Smart BIT software when a scenario is run.

The INDICATOR-WINDOW flavor object provides an indicator "lamp"-type display for the BIT status window. Indicators may display any allowable color in the color registry table. When an indicator window object instance is created, it is initialized by first creating and clearing a bitmap of the chosen size and color; then it is displayed with its window title. Data structures (lists) within the flavor object for each indicator are then set up, the title labels for each indicator parameter are displayed, and the indicator is drawn with the default color. The change of indicator display color is done by calling the appropriate INDICATOR-WINDOW method (object function) with the selected indicator "lamp" and the selected color as formal parameters.

The BIT status window is an instance of the INDICATOR-WINDOW flavor object. It displays the following labels with a status indicator for each: "LRU BIT Report Status" (LRU BIT report), an "OFP BIT Status" (an assessment of the LRU BIT based on how a typical modern aircraft Mission Computer OFP would react to an LRU BIT fault report), and the "Smart BIT Status" (Smart BIT assessment of the LRU BIT). The indicators have the appearance of indicator lamps that display NO-GO (red) or GO (green) each time the Smart BIT software reads in a message from the LRU BIT and TSMD data. The line graph windows for the LRU BIT, Adaptive BIT, and Temporal Monitor therefore display a historical record of the scenario while the BIT Status window indicators provide the current status and assessment of the scenario data as they are being read in and evaluated by the Smart BIT software. These windows allow the observation of the Smart BIT AI software performance in reducing the false alarm reports of a simulated LRU that is exhibiting faulty behavior.

# 4 – INTEGRATION DESIGN CRITERIA

The Smart BIT/TSMD Integration System unifies the BIT and environmental stress simulation, LRU simulation, time stress measurement, and Smart BIT functions into a single test bed. In order to achieve this capability, certain integration issues had to be considered during the design phase. Topics such as how fast data are transferred around the Smart BIT/TSMD Integration System, what types of information must be passed to and from each computer subsystem, and how information that is passed between each computer is integrated into its internal processing are provided in the following discussion.

The selection of the rate at which data are transferred around the Smart BIT/TSMD integration system was based on actual avionic LRU rates, the limitations of IBM PC AT computers, how much data the Smart BIT and TSMD computers could process, store, and display, and how long a scenario should run.

The SCADC was chosen as the avionic LRU that is to be simulated by the LRU BIT computer. This LRU usually transfers its data at a 16-Hz rate. However, IBM PC AT-type computers cannot set up to interrupt at this 16-Hz rate due to limitations in the internal IBM PC AT timer. Therefore, the closest rate possible for an IBM PC AT-type computer is about 18 Hz. It should be noted that it is extremely difficult for application software written for a multitasking operating system to maintain absolute control over message output time intervals. For instance, it has been observed that Microsoft Windows periodically may "steal" an 1/18th-second interval from the UUT computer application program to do internal housekeeping functions. Similarly, the Smart BIT computer (18 times/second) message timing (between the Smart BIT and LRU BIT computers as well as the Smart BIT and TSMD computers) may be disrupted temporarily for ephemeral garbage collection or checking for mouse/keyboard entry within the Common LISP environment, as well as Unix System V and X11 Window task swapping. As a result, message traffic frequencies are close to the indicated rates, but not always exact.

Another issue to consider was how long a scenario should run. This decision was based on Statement Of Work requirements that:

"The time scale shall be compressed so as to provide a wide variety of conditions in a relatively short period of time."

It was also based on the premise that the Smart BIT/TSMD integration system is to be used as a demonstration tool. A maximum interval of one minute (60 seconds) was chosen because it meets the Statement Of Work requirement, the demo tool requirement, and the fact that transferring data to and from the LRU BIT computer and the Smart BIT computer at about 18 Hz for close to a minute provides a large enough data set (about 700 samples) for both the Smart BIT and TSMD computers to perform their functional tasks as well as demonstrate their capabilities. (This does not mean that 700 samples are necessary in order to show Smart BIT and TSMD techniques. It just represents a typical data set size.)

The type and format of the data that are transferred to and from the Smart BIT and TSMD computers were based on the Smart BIT / TSMD integration Statement Of Work and on previous work done by Westinghouse on the Micro-TSMD project. The environmental stress information transmitted to the Smart BIT computer by the TSMD computer includes voltage, temperature, and vibration data. The format of these data is based on the Westinghouse Micro-TSMD project, and the type of sensors, as well as on the scaling of the analog-to-digital (A/D) converters on the National Instruments multifunction (AT-MIO-16H) card that resides in the TSMD computer. The Smart BIT computer takes this environmental stress information and combines it with the LRU data for the purpose of trying to "learn" any associations between the environmental stress data and the LRU BIT data. The integration of environmental stress and the LRU BIT information within the Smart BIT computer software required some reformatting and rescaling of both data types in order for proper processing to occur. The rescaling of these data is accomplished in part by a LISP definition function table. The information in this function table tells other LISP software what the scaling is for each data word that is read in from the LRU BIT computer (via MIL-STD-1553B bus) and the TSMD computer (via IEEE-488 bus). In addition, LISP macro functions are created to reformat the data to allow the Neural Network, K-Nearest Neighbor, and Temporal Monitor functions to properly process this information.

The digital communication hardware bus protocols that were used in the Smart BIT/TSMD integration system were industry standards (standard bus communication protocols used: RS-232C, MIL-STD-1553B, and IEEE-488). Therefore, all digital communication hardware handshaking, was taken care of by commercial-off-the-shelf hardware. However, the actual message format and software protocol used were based partly on the Westinghouse Micro-TSMD project. This protocol was built upon to allow the implementation of a bus master/slave type control mechanism for two computers (for a more complete discussion of this protocol see sections 6.2 and 6.3 of the Interface Design Document for this effort). This approach allowed a faster implementation of the integration of TSMD computer data with the Smart BIT computer (via the IEEE-488 bus) because it was based on previously successful TSMD communication efforts. In order to maintain consistency of software and a high level of reliability and data integrity, similar message formats and software protocols were used for the UUT computer and LRU BIT computer RS-232C interface.

During a scenario, the Smart BIT computer transmits the LRU BIT word to the TSMD computer. This LRU BIT information is saved with the environmental stress data into TSMD computer's long-term memory. The combination of environmental stress and LRU BIT information may be analyzed off-line (not real time) to uncover other potential correlations between environmental stress data and LRU BIT data. The format and content of these data messages for the Smart BIT and TSMD computer are specified in the Interface Design Document for this effort. Both the transmission of the LRU BIT word by the Smart BIT computer to the TSMD computer and the environmental stress data from the TSMD computer to the Smart BIT computer are constantly occurring (in real time) while the scenario continues to run. This real-time exchange of information between the Smart BIT computer and the TSMD computer more tightly binds and enhances the Smart BIT and TSMD techniques.

TSMD event processing techniques are primarily intended for collecting data for off-line analysis (non-real-time type functions). Smart BIT techniques require more timely, real-time information. This apparent paradox has precipitated an adjustment in the rate that the TSMD computer samples the environmental sensor data and reports its information to the Smart BIT computer. Typical TSMD techniques require data to be read in at about a one-second interval and reported out at very long time intervals (perhaps weeks). Since the Smart BIT computer must correlate this environmental stress data with LRU BIT messages that occur at much faster rates, both the scanning of the environmental sensor data and the transfer rate (to the Smart BIT

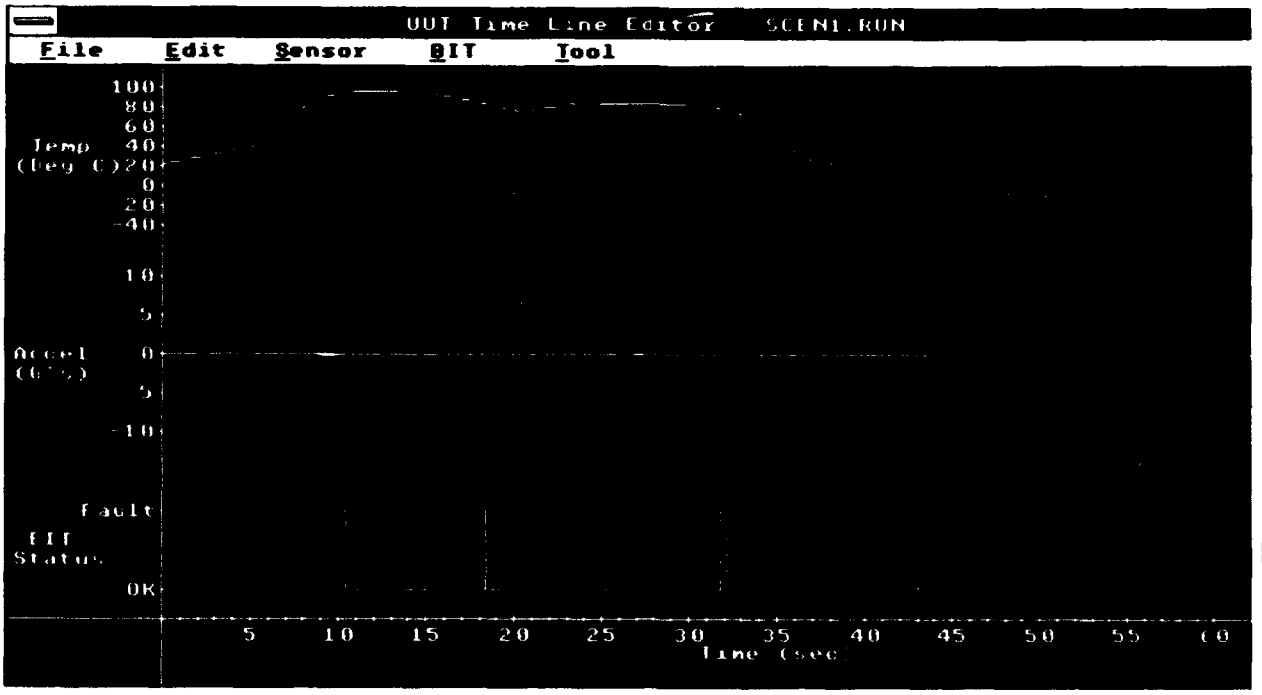
computer) have been increased to 10 times/second and 18 times/second, respectively. Ideally, both rates should be about the same. However, the TSMD computer has a fairly high processing burden of displaying information in a number of windows in addition to performing other tasks. Thus, increasing the input data sample rate to 18 times/second might cause the TSMD computer software to lose most or all of any spare real-time reserves. This compromise is effective because variations in temperature are quite slow relative to the TSMD sample rate (reading in environmental sensor data) and because much of the vibration data (vibration peak, integral, etc) is processed by the Westinghouse signal conditioning circuitry (on the sensor card) hardware during the interval sample time.

# 5 – DEMONSTRATION SCENARIOS

This section presents three demonstration scenarios for the Smart BIT/TSMD Integration System. The LRU chosen for the three scenarios delivered with the Smart BIT/TSMD Integration system is a SCADC. All fault scenarios were obtained from the SCADC FMECA reports (Failure Mode, Effect, and Criticality Analysis Report for the SCADC CPU-142/A). The three scenarios illustrate Smart BIT/TSMD Integration techniques by providing three distinct temperature, vibration, and BIT time lines with three distinct LRU scenario data sets that represent three different SCADC LRU FMECA failure modes (usage of the term "BIT report" or "SCADC BIT" refers to a partial simulation of the SCADC LRU and its BIT, running on the LRU BIT computer for these three scenarios). Each scenario requires four different UUT computer files to run. They include an environmental stress time line file (e.g., SCEN1.TSM - contains temperature and accelerometer time line information), a BIT time line file (e.g., SCEN1.BIT), an LRU parametric scenario file (e.g., SCEN1.DAT), and a run file (e.g., SCEN1.RUN). (All scenario no. 1 files are named SCEN1, all scenario no. 2 files are named SCEN2, and all scenario no. 3 files are named SCEN3; it is the file extent (for example: BIT) that determines the type of file.) All three scenarios used the UUT computer to simulate the temperature and accelerometer sensor signal output (instead of the physical sensors) because repeatability of results was necessary for the final demonstration. The following discussion explains each scenario and its affect on all computer subsystems.

## 5.1 SCENARIO NO. 1

The first scenario contains a temperature stress component with three sets of BIT fault bursts during the high-temperature intervals (see the UUT computer time line for scenario no. 1, Figure 28). The temperature time line for scenario no. 1 contains relatively rapid changes as well as two areas of sustained high-temperature values. This type of profile might be observed in an aircraft that is performing a type of maneuver that causes the LRU's cool air flow to be cut off, and then restored after the maneuver has completed. Scenario no. 1 contains only the temperature stress profile with no accelerometer activity. The BIT time line editor (labeled: BIT Status on the UUT computer BIT time line editor area) determines when BIT faults are to be reported by the LRU BIT computer to the Smart BIT computer. Scenario no. 1 has three sets of BIT fault reports.




R91-6080-030

**Figure 28. UUT computer time line for scenario no. 1.**

Each set is a burst of consecutive BIT faults. They occur at the 10.5-, 18.5-, and 31.8-second points in the scenario. The UUT computer's BIT status time line editor also controls the length of the scenario. The total time for the first scenario is therefore about 35.5 seconds. This time interval is slightly shorter than the temperature profile. Therefore, the temperature at the end of the scenario, as controlled by the UUT computer BIT time line, is about at the 50°C level.

The LRU BIT and parametric information that describes a SCADC LRU for scenario no. 1 is located in the SCEN1.DAT file (see Table 3 for the contents of this file). The RT No. and RT subaddress, as well as the format and resolution of the MIL-STD-1553B message words presented in the SCEN1.DAT scenario no. 1 file, were taken from the SCADC interface control document (GASD ICD-SCADC-1553). Scenario no. 1 illustrates the "All Outputs Fixed" FMECA fault scenario in which the pressure altitude word reflects a fixed-to-zero fault value and the SCADC BIT word indicates the faulted area (pressure altitude is invalid). All data in the SCEN1.DAT file are shown in decimal format. When scenario no. 1 begins, the LRU BIT computer reports the information in the initial value column (in the SCEN1.DAT file, Table 3) to the Smart BIT computer. As the scenario progresses, the corresponding parametric values in the

increment value column are added periodically to the parametric values in the initial value column until the parameter reaches the value specified in the final value column. This simulates the dynamic conditions that would be present if the SCADC were "flying" in the aircraft. For instance, the true airspeed SCADC parameter increments from 350 knots initially to a final value of 540 knots with an increment of 2 knots. The fault value column specifies the parameter value when a fault occurs. Each parameter having a different initial value and fault value is then set to the fault value data whenever a BIT fault occurs. For instance,when the BIT word (word no. 1) is reporting OK, its decimal value sent by the LRU BIT computer to the Smart BIT computer is specified in the initial value column (65280 decimal which is FF00 in hexadecimal). When a BIT fault report occurs, the LRU BIT computer reports a decimal value of 31488 or 7B00 hexadecimal (from the Fault Value column) to the Smart BIT computer . This flexible approach allows a change to a different LRU (instead of the SCADC) to be easily accomplished for the UUT computer by changing the MIL-STD-1553B message word data from the new LRU's interface control document and supplying it in the format shown for the SCEN1.DAT file (changing to a new LRU also requires changes that are discussed in the programmer's manuals to the LRU BIT computer and the Smart BIT computer programs).

At the beginning of the scenario, the UUT computer sets the simulated LRU power to ON. This signal passes through the TSMD sensor card (residing in the LRU BIT computer chassis) to indicate to the TSMD computer that the LRU is active and the scenario is beginning (the power signal also is sent to the Smart BIT computer by the TSMD computer). The TSMD computer reports this LRU power-on event in the event buffer data window (see bottom window of the TSMD computer display for scenario no. 1, Figure 29). During the course of the scenario, the TSMD short-term data window continually displays the changes in temperature versus time. The temperature data are output by the UUT computer according the scenario no. 1 temperature profile (UUT computer temperature time line, Figure 28). As the temperature passes through a threshold (shown as horizontal dotted lines in the TSMD short-term data window) a temperature event, its bin number, and the approximate time the temperature event crossed the bin threshold is displayed in the event buffer data window (see bottom window of the TSMD computer display for scenario no. 1, Figure 29). The bottom threshold of the TSMD short-term data window is referred to as bin 0 and encompasses the -40°C to approximately -30°C temperature range. The next bin is referred to as bin 1, and it represents the temperature range of -29 to about -5°C. Bin 2 is -4 to 18°C, bin 3 is 19 to 40°C, bin 4 is 41 to about 65°C, and bin 5 is approximately 66 to

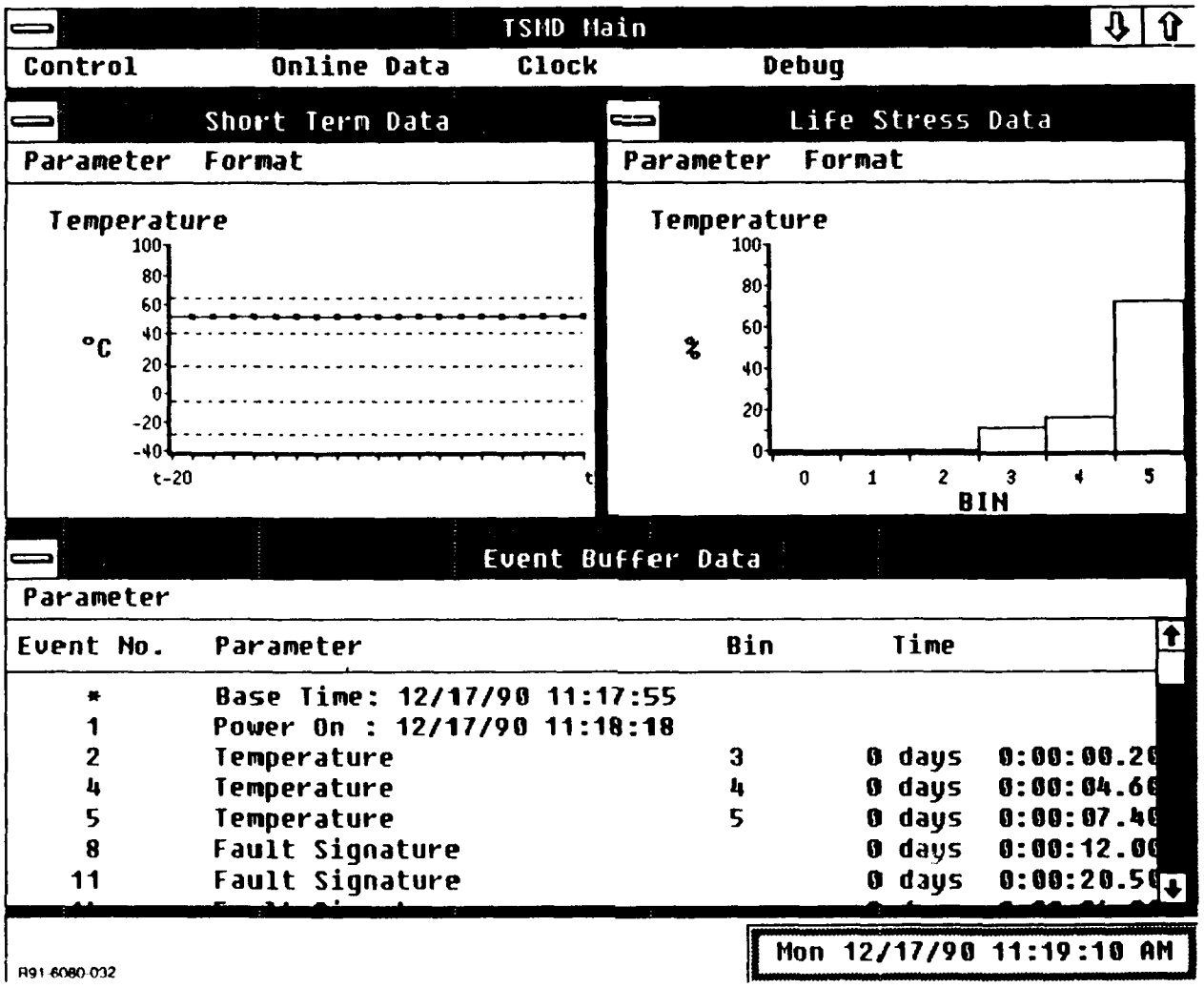



Figure 29. TSMD computer display for scenario no. 1.

100°C. The relative time (as a percentage of total scenario time) that the temperature remains in a particular threshold area (bin) is shown in the life stress data window (see upper right window of the TSMD computer display for scenario no. 1, Figure 29). The life stress data window indicates that the temperature stayed in bin 5 (bin 5 temperature range: about 66 to 100°C) approximately 73% of the time during scenario no. 1. This agrees with the UUT computer temperature time line shown in Figure 28. The temperature at the end of the scenario is shown to be at about the 50°C level (see the short term data window shown in the top left of the TSMD computer display for scenario no. 1, Figure 29). This agrees with the expected temperature value from the UUT computer at the end of scenario no. 1. The event buffer data window also indicates BIT fault report events during the scenario. This information reflects the approximate

time that an BIT fault is reported. The TSMD BIT fault event time is somewhat delayed from the actual time shown on the UUT BIT time line because the BIT fault information passes from the UUT, through the LRU BIT, Smart BIT, and finally the TSMD computer before it is logged into the TSMD computer event buffer data window.

The Smart BIT computer reads the simulated SCADC BIT and parameter scenario data (defined in the UUT computer scenario generator editor scenario no. 1) from the LRU BIT computer via the MIL-STD-1553B bus periodically (about 18 times/second). The Smart BIT computer also reads in the environmental data from the TSMD computer (via the IEEE-488 bus) periodically (about 18 times/second). These data are combined, and then reformatted and rescaled for compatibility with the Smart BIT methods. The information is then analyzed by the Temporal Monitor and either the Neural Network or K-Nearest Neighbor Adaptive BIT techniques are displayed on the Smart BIT computer screen. The Smart BIT computer displays separate line graph windows for the UUT BIT Test (the LRU BIT report), the output of the Neural Network, the output of the Temporal Monitor, and the Temporal Monitor Transition probabilities. In addition, the Smart BIT computer screen includes a BIT status window and an interactive text display called a LISP LISTENER window.

The Smart BIT computer UUT BIT Test window for scenario no. 1 displays the simulated SCADC BIT report data for scenario no. 1 (see the Smart BIT computer display for scenario no. 1, Figure 30). It is similar to the UUT computer BIT time line editor for scenario no. 1 (see the UUT computer time line for scenario no. 1, Figure 28). It should be noted that the UUT computer and the Smart BIT computer monitor screens are not the same resolution (the Smart BIT computer is a much higher resolution than the UUT computer) and, therefore, the data displayed may appear to be slightly different. The Smart BIT Neural Network Adaptive BIT classification window reflects the output of the neural network. While scenario no. 1 is running, this window continually updates the Neural Network output as more input data message samples (LRU and TSMD data) are read in. At first, the Neural Network output classification flips from the Inter. (intermittent) to OK and back as the Neural Network attempts to minimize its average error. After several training cycles, the BIT and TSMD environmental stress data associations are "learned" by the Neural Network. This occurs when the data in the Neural Network Adaptive BIT classification window mimic the data shown in the UUT BIT test window (see the Smart BIT computer display for scenario no. 1, Figure 30).

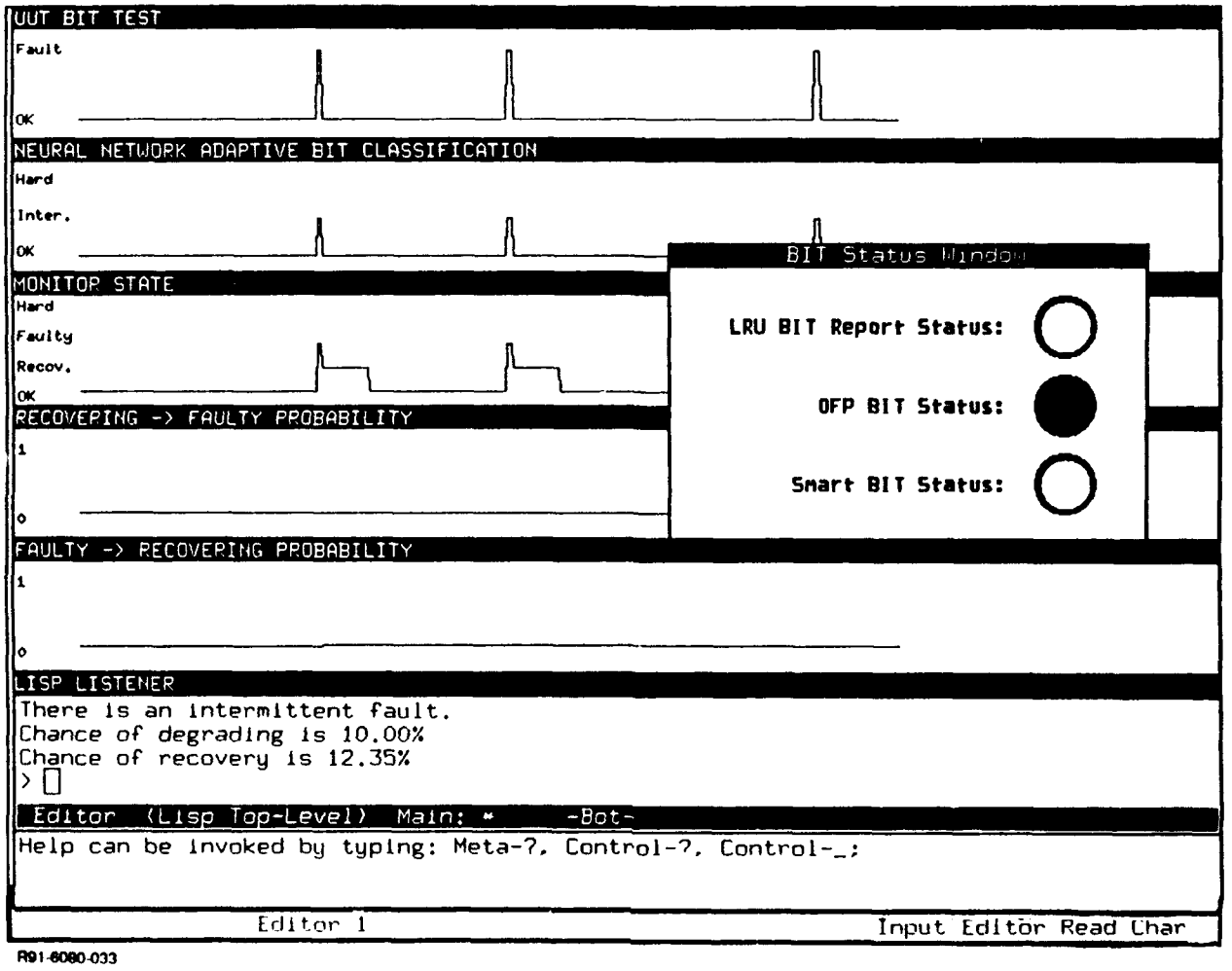



**Figure 30. Smart BIT computer display for scenario no. 1.**

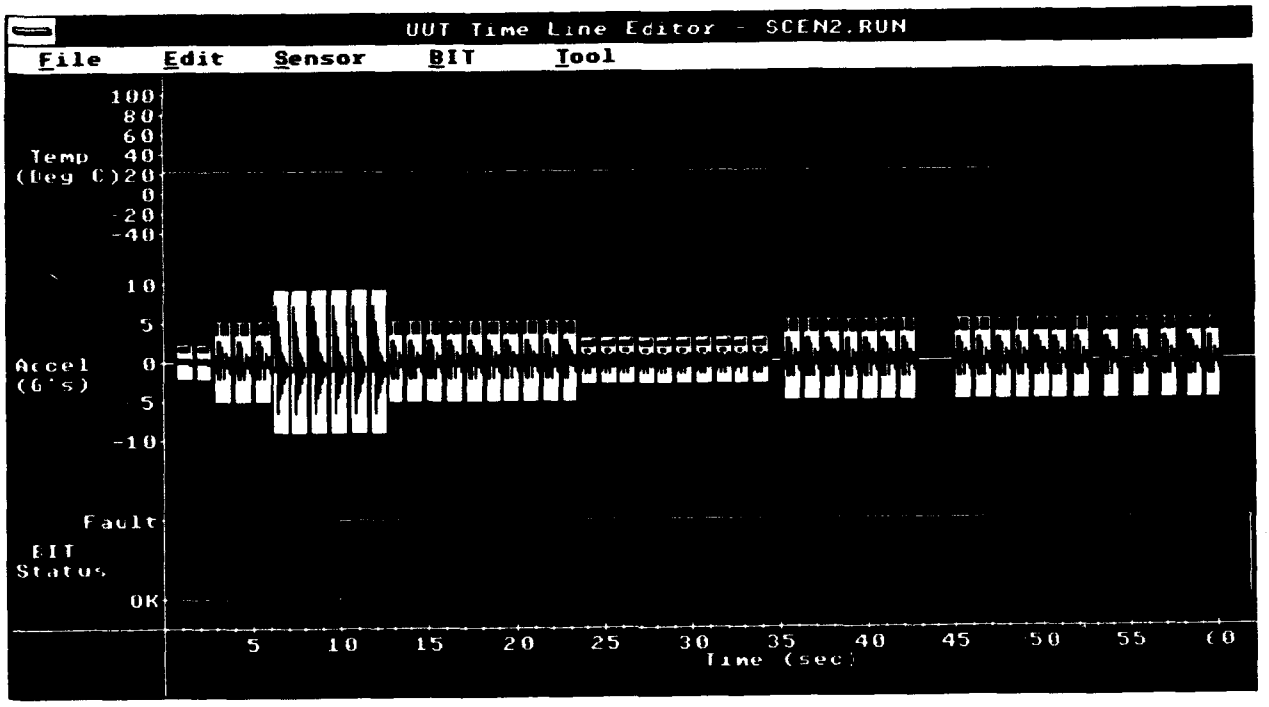
The Temporal Monitor views the same BIT data as the Neural Network, but attempts to classify the input data according to its past history. When an BIT fault report is encountered, the Temporal Monitor changes its state to Intermittent (Faulty). It remains in this state for each BIT fault report. Once the BIT report returns to an OK condition, the Temporal Monitor changes its state to Recov. (Recovering). It remains in this state for successive OK BIT reports until the transition probabilities indicate that it is time to return to the OK state. The Recovering to Faulty, and Faulty to Recovering probabilities are shown in the two line graph windows below the Temporal Monitor state window (see the Smart BIT computer display for scenario no. 1, Figure 30). The Smart BIT Temporal Monitor data for scenario no. 1 indicates that small numbers of BIT transitions result in small changes in the transition probabilities.

The BIT status window contains three "lamp" indicators that change color (green for OK and red for Fault) according to the input data. (see the BIT status window in the Smart BIT computer display scenario no. 1, Figure 30). The BIT status window shown in Figure 30 for scenario no. 1, depicts the "lamp" indicators as an empty circle when the display screen indicator is actually green, and shows a black-filled circle when the "lamp" indicator on the actual Smart BIT display screen is red. The LRU BIT report status displays the state of the current BIT report message. It therefore changes color according to the data in the UUT BIT Test window (changes to red for a BIT fault, and green for an OK BIT message). The OFP BIT status reflects the state of the Mission Computer's Operational Flight Program (OFP) when a BIT fault report is received. When a BIT fault report is recognized by an actual Mission Computer's OFP, it assumes a hard fault condition and "reconfigures" to try to use different equations while running in a degraded state. This type of status is represented by the OFP BIT status indicator as a red color for a fault condition. The Smart BIT status "lamp"indicator displays the current Smart BIT assessment of the LRU (green for OK, red for Fault) for each BIT message that is processed.

The LISP LISTENER window prints out the Smart BIT assessment for the scenario no. 1 data. It states correctly that an intermittent fault has occurred, the chance that the LRU may become intermittent is about 10.00%, and the chance for the LRU to remain OK is about 12.35%. These probabilities may be different if scenario no. 1 is run again, because the Smart BIT/TSMD Integration System test bed contains four asynchronously running computers with different latency times to process messages. For example, some BIT faults that the UUT computer commanded the LRU BIT computer to report might occasionally be missed by the Smart BIT computer. This turns out to be more reflective of actual avionic systems than having all computers synchronously locked to each other.

## 5.2 SCENARIO NO. 2

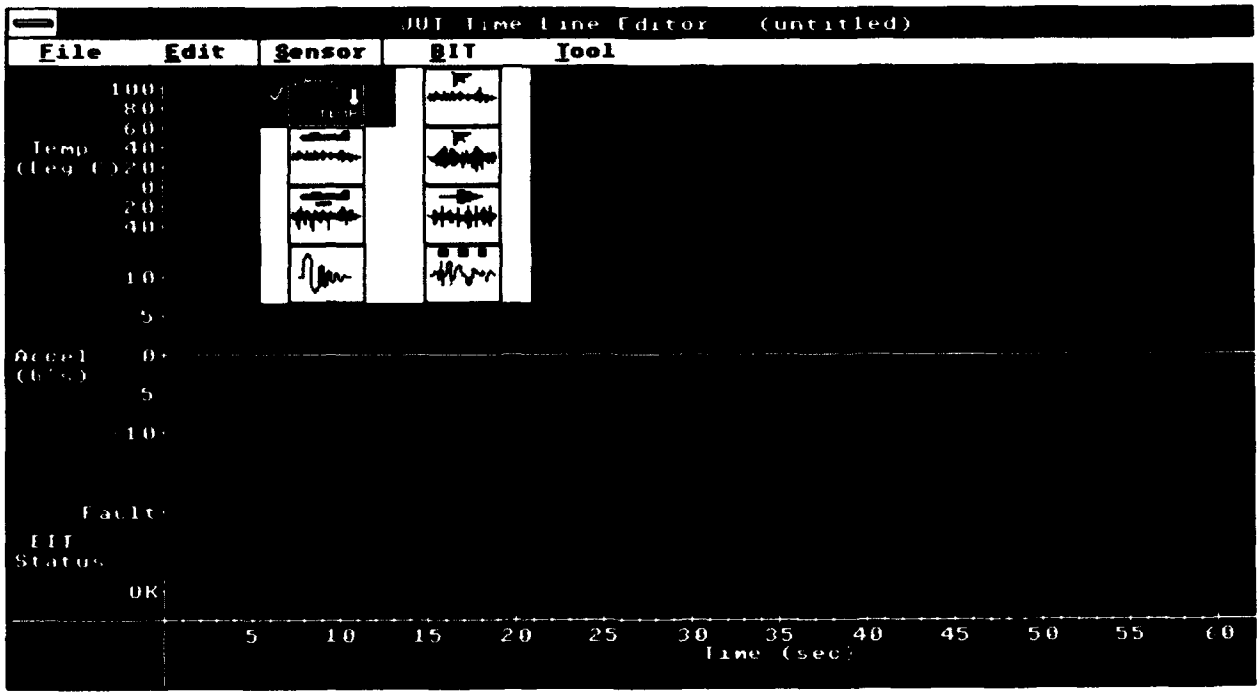
The second scenario contains an accelerometer stress component with BIT faults that begin at the 10-second point in the scenario and continue for the duration (about 50 seconds) of the scenario time (see the UUT computer time line for scenario no. 2, Figure 31). This represents a hard fault condition that may have been caused by the large vibration activity. Scenario no. 2 maintains a nominal temperature value (approximately 22°C) for the entire scenario time. The accelerometer waveforms that are shown as icons in the accelerometer time line editor are based on actual flight data that were translated into the time domain from the spectral density plots (see Subsection 2.1, Aircraft Environmental Factors, for a more detailed discussion of these data).




R91-6080-034

**Figure 31. UUT computer time line for scenario no. 2.**

The sensor menu of the UUT computer time line editor contains seven different accelerometer waveforms that can be selected. The accelerometer sensor menu (shown in Figure 32) options provide both an iconic representation of the waveform itself as well as an iconic depiction of the flight conditions when the data were obtained. For instance, the second (refer to Figure 32) icon down in the top left column represents the acceleration for level flight conditions. The icon just below level flight represents a weapons drop; the next is a damped sinusoid which might represent an external impact to the aircraft. The first icon on the top right column is a level turn, the one below it is a maximum g turn, the third from the top right represents an aircraft side slip, and the bottom right icon represents speed brakes being applied. All of the icons in the accelerometer time line editor are of a one-second duration. The vertical size of the icon approximates the peak value of the g force. For instance, the damped sinusoid (largest of the icons) has a peak value of about ± 10 g. The first accelerometer icons in scenario no. 2 (extreme left in the UUT computer accelerometer time line for scenario no. 2, Figure 31) represents level flight. The next set of three icons (moving to the right in the accelerometer time line), represents speed brakes being applied. The damped sinusoid (the next six icons) might represent an external impact on the aircraft. Speed brakes are shown being applied again (for the next ten icons), followed by a weapons drop (the next ten icons), and finally, speed brakes for the




R91-6080-035

**Figure 32. UUT computer time line editor sensor menu options.**

duration of the scenario. Accelerometer icons may be located close (but may not overlap) to each other to allow the waveform to be repeated and thus obtain the desired time duration beyond the one-second time interval of a single accelerometer waveform.

The BIT and parametric information that describes a SCADC LRU for scenario no. 2 is located in the SCEN2.DAT file (see Table 4 for the contents of the SCEN2.DAT file). The RT No. and RT subaddress as well as the format and resolution of the MIL-STD-1553B message words presented in the SCEN2.DAT scenario no. 2 file were taken from the SCADC interface control document (GASD ICD-SCADC-1553). Scenario no. 2 illustrates the "No Indicated Airspeed (Coarse)" FMECA fault scenario in which the true airspeed word is set to a fault value of zero and the SCADC BIT word indicates the faulted area (TAS/air density ratio, and calibrated airspeed invalid). All data in the SCEN2.DAT file are shown in decimal format. When scenario no. 2 begins, the LRU BIT computer reports the information in the initial value column to the Smart BIT computer (see Table 4). As the scenario progresses, the corresponding parametric values in the increment value column are added periodically to the parametric values in the initial value column until the parameter reaches the value specified in the final value column. This simulates the dynamic conditions that would be present if the SCADC were

Table 4. Scenario No. 2 (SCEN2. DAT File)

; SCENARIO NAME: NO INDICATED AIRSPEED (COARSE)

;**********

; NEW MESSAGE:

; COMMAND IS SCADC MAIN OUTPUT MESSAGE:

; RT # RT SUBADDRESS
28 16

; MODE CODE # MODE CODE DATA
99 99

| ; WORD # | DESCRIPTION | UNITS | DATA TYPE | INITIAL VALUE | INCREMENT VALUE | FAULT VALUE | FINAL VALUE | RESOLUTION | LSB BIT # |
|---|---|---|---|---|---|---|---|---|---|
| 1 | BIT_WD | NONE | BOOLEAN | 65280.0 | 0.0 | 55040.0 | 65280.0 | 1.0 | 16.0 |
| 2 | PRES_ALT | FT | SIGNED | 10548.0 | 10.0 | 10548.0 | 12288.0 | 2.5 | 16.0 |
| 3 | BARO_ALT | FT | SIGNED | 10548.0 | 10.0 | 10548.0 | 12288.0 | 2.5 | 16.0 |
| 4 | TRUE_AIRSPEED | KNOTS | SIGNED | 532.0 | 1.0 | 0.0 | 634.0 | 0.125 | 15.0 |

MR91-6080-036

"flying" in the aircraft. For instance, the true airspeed SCADC parameter increments from 532 knots initially to a final value of 634 knots with an increment of 1 knot. Similarly, the pressure and baroset altitude data words initially start at 10548.0 ft, and increment by 10 ft to a final value of 12288.0 ft. The fault value column specifies the parameter value when a fault occurs. Each parameter that has a different initial value and fault value is then set to the fault value data whenever a BIT fault occurs. For instance,when the BIT word (word no. 1) is reporting no errors, its decimal value that is sent by the LRU BIT computer to the Smart BIT computer is specified in the initial value column (65280 decimal which is FF00 in hexadecimal). When an BIT fault report occurs, the LRU BIT computer reports a decimal value of 55040 to the Smart BIT computer, which is D700 in hexadecimal (taken from the fault value column).

As described earlier, the UUT computer sets the simulated LRU power to ON at the beginning of the scenario. The TSMD computer reports this LRU power-on event in the event buffer data window (see bottom window of the TSMD computer display for scenario no. 2, Figure 33). During the course of the scenario, the short term data window continually displays the changes in vibration peak data versus time. This vibration peak data are output by the UUT computer according to the scenario no. 2 accelerometer time line profile (see UUT computer accelerometer time line, Figure 31). As the vibration peak value passes through a threshold (shown as horizontal dotted lines in the TSMD short-term data window) a vibration peak event, its bin number, and the approximate time the vibration peak event crossed the bin threshold is displayed in the event buffer data window (see bottom window of the TSMD computer display

for scenario no. 2, Figure 33). The bottom threshold of the TSMD short-term data window is referred to as bin 0 and encompasses the 0 to approximately 2-g vibration peak range. The next bin is referred to as bin 1 and represents the vibration peak range of 3 to about 6 g. Bin 2 contains the range of 7 to 10 g, bin 3 is 11 to 15 g, bin 4 is 16 to about 18 g, and bin 5 is approximately 19 to 25 g. The relative time (as a percentage of total scenario time) that the vibration peak remains in a particular threshold area (bin) is shown in the life stress data window (see upper right window of the TSMD computer display for scenario no. 2, Figure 33). The life stress data window indicates that the vibration peak stayed in bin 1 approximately (bin 1 range: about 3 to 6 g) 73% of the time during scenario no. 2. This agrees with the UUT computer accelerometer time line shown in Figure 31 because most of the accelerometer waveforms in

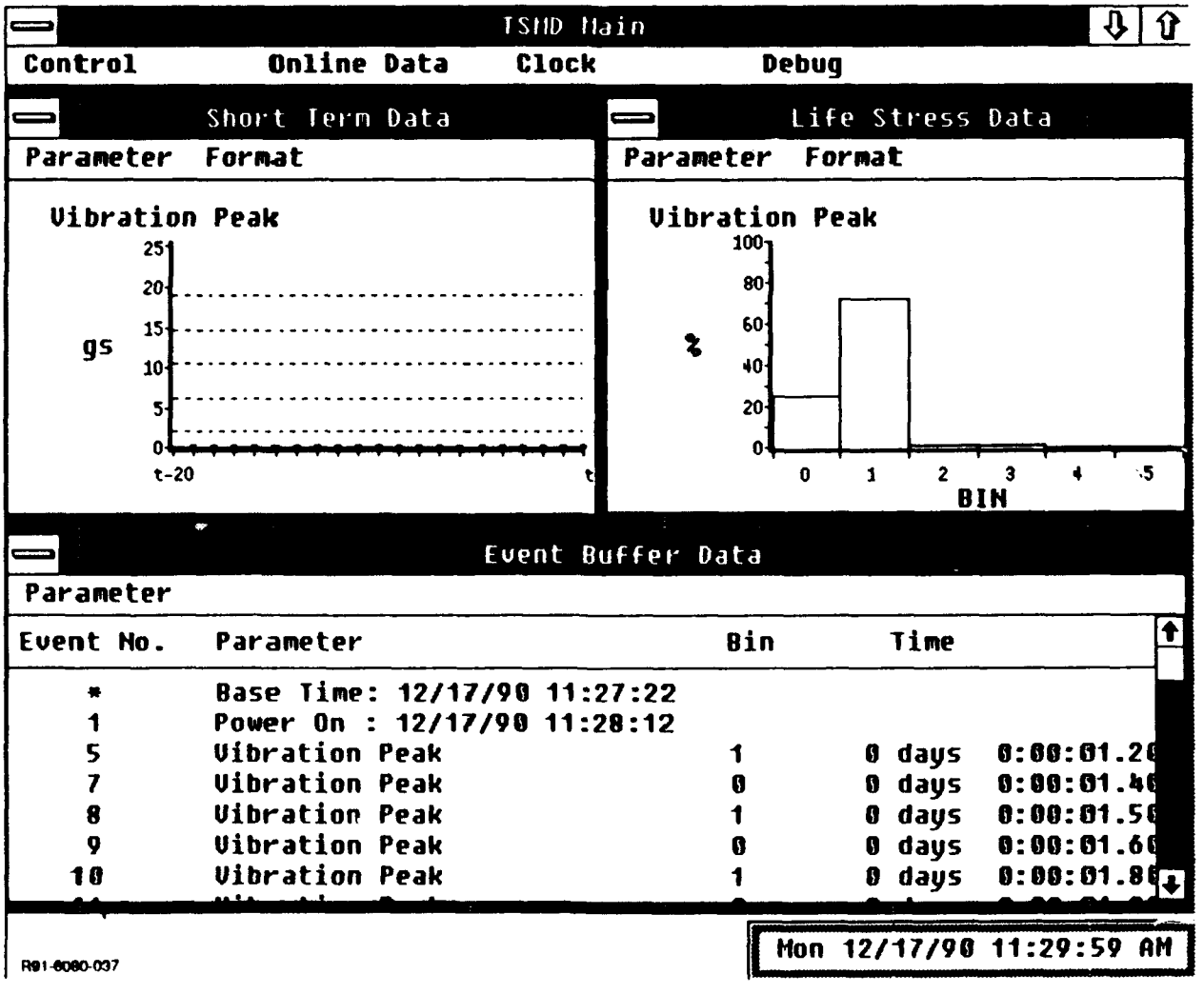
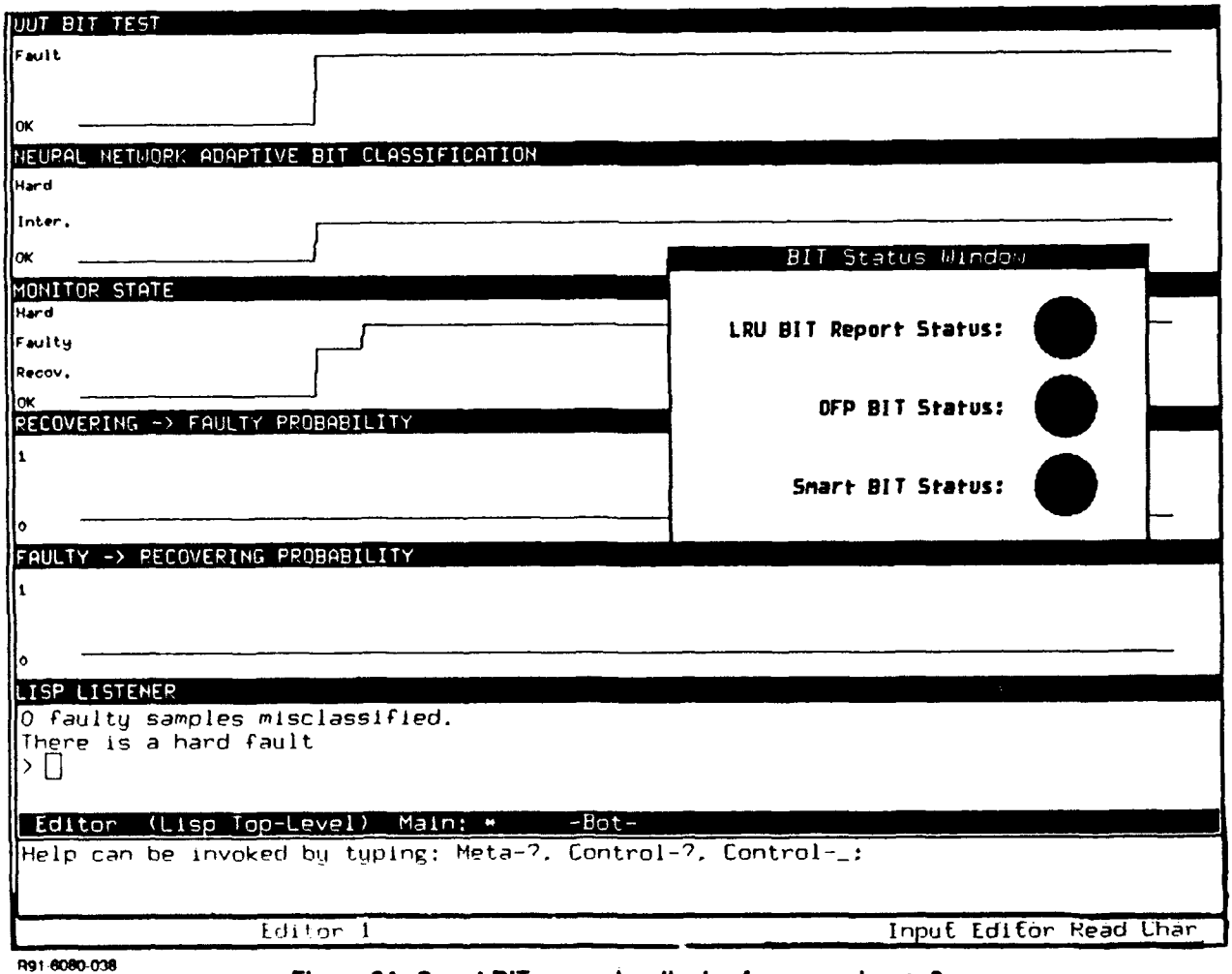



Figure 33. TSMD computer display for scenario no. 2.

scenario no. 2 have a peak g force value of between 3 and 5 g. Most of the rest of the vibration peak data remained in bin 0 (0 - 2 g) 25% of the total scenario time. This also correlates with the UUT computer accelerometer time line shown in Figure 31 due to some intervals of no accelerometer activity in the time line, and the fact that the level flight and weapons drop accelerometer waveforms contain peak g forces between 2 and 3 g. The damped sinusoid waveforms do not contribute much to the overall life stress data (about 2% in bins 2 and 3) because the 10-g peak vibration force is not sustained for very long due to the speed of the decaying sinusoid amplitude.

The Smart BIT computer UUT BIT Test window for scenario no. 2 displays the simulated SCADC BIT report data for scenario no. 2 (see the Smart BIT computer display for scenario no. 2, Figure 34). It is similar to the UUT computer BIT time line editor for scenario no. 2 (see the




R91-6080-038

**Figure 34. Smart BIT computer display for scenario no. 2.**

UUT computer time line for scenario no. 2, Figure 31). The Smart BIT Neural Network adaptive BIT classification window reflects the output of the neural network. While scenario no. 2 is running, this window continually updates the Neural Network output as more input data message samples (BIT and TSMD data) are read in. At first, the output classification flips from the Inter. (intermittent) to OK and back as the Neural Network attempts to minimize its average error. After several training cycles, the BIT and TSMD environmental stress data associations are "learned" by the Neural Network. This occurs when the data in the Neural Network Adaptive BIT classification window mimic the data in the UUT BIT test window (see the Smart BIT computer display for scenario no. 2, Figure 34).

The Temporal Monitor analyzes the same BIT data as the Neural Network, but attempts to classify the input data according to their past history. When an BIT fault report is encountered, the Temporal Monitor changes its state to Intermittent (Faulty). It remains in this state for each consecutive BIT fault report until enough of them exceeds a dynamically set threshold. The Temporal Monitor then transitions to the Hard state. Since no OK BIT reports occur after the start of the BIT fault reports, the Temporal Monitor remains in the Hard state (see the Smart BIT computer display for scenario no. 2, Figure 34). The Smart BIT Temporal Monitor data for scenario no. 2 indicates that when only one transition (BIT OK to BIT fault) occurs,the transition probabilities experience little or no change.

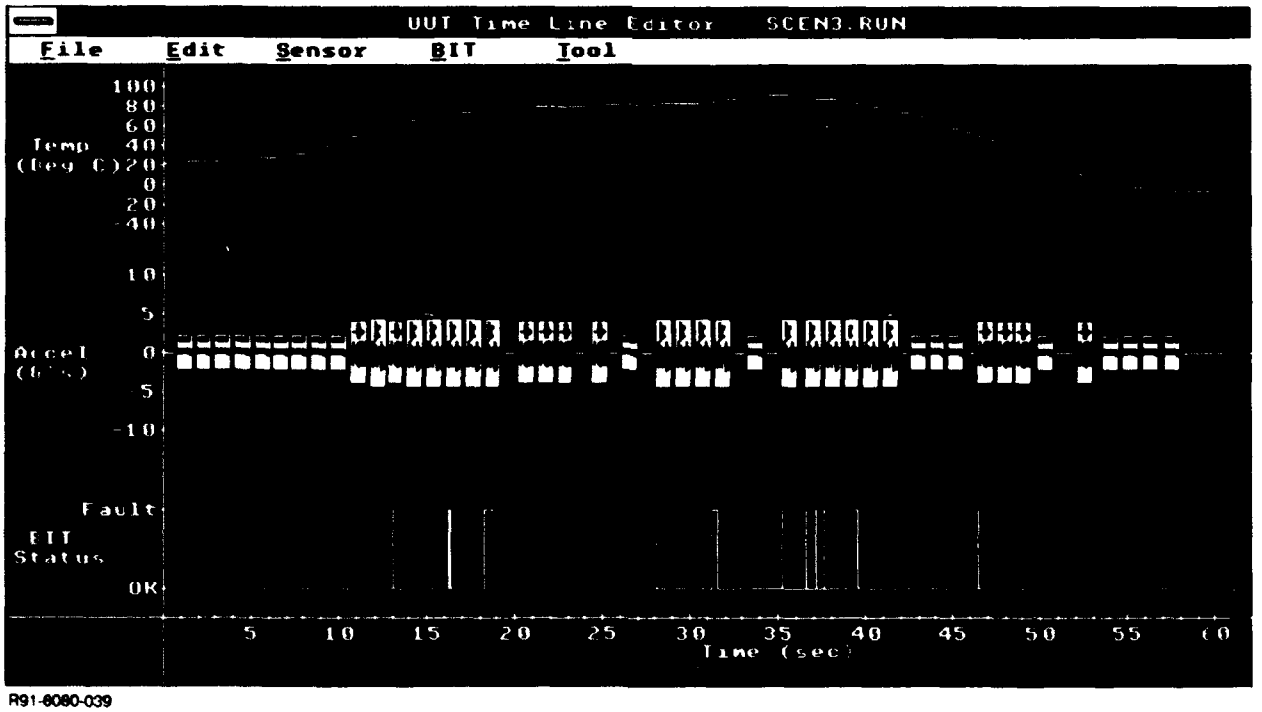
The BIT status window contains three "lamp" indicators that change color (green for OK and red for Fault) according to the input data (see the BIT Status Window in the Smart BIT computer display for scenario no. 2, Figure 34). The BIT status window in the Smart BIT computer display for scenario no. 2, Figure 34, shows that all of the "lamp" indicators are black-filled circles and therefore all BIT status (actually displayed on the Smart BIT computer screen) indicators are showing the red color (fault). The LRU BIT report status indicator displays the state of the current LRU BIT report message (in this case it shows a fault because the last message data contained a BIT fault. However, at the beginning of the scenario, this indicator displayed the OK BIT state that occurred in the first 10 seconds of the BIT data). It therefore changed to the red (fault) color when the first BIT fault data was received. The OFP BIT status reflects the state of the Mission Computer's OFP when an BIT fault report is received. The OFP BIT status indicator also shows a red color (black-filled circle shown in Figure 34) for a fault condition, reflecting the fact that an actual Mission Computer's OFP would assume (correctly in this case) that a hard fault condition has occurred and would "reconfigure" to try to run in a

degraded state if it was presented with a similar scenario. The Smart BIT status "lamp" indicator displays the current Smart BIT assessment of the LRU (in this case, red for Fault) for each BIT message that is processed. This is shown in Figure 34 as a black-filled circle. Therefore, all indicators correctly show a hard fault condition.

The LISP LISTENER window prints out the Smart BIT assessment for the scenario no. 2 data. It states correctly that a hard fault condition has occurred. It also indicates that the Neural Network properly classified all LRU BIT and TSMD data samples.

## 5.3 SCENARIO NO. 3

The third scenario contains temperature and accelerometer stress components with several sets of single impulse and burst BIT faults (see the UUT computer time line for scenario no. 3, Figure 35). The temperature time line for scenario no. 3 contains a relatively long interval of sustained high-temperature values. This type of profile might be observed in an aircraft when the cool air flow to the LRU is gradually cut off, then gradually restored during the mission. The accelerometer activity for scenario no. 3 includes a combination of level flight, side slip, and max g turns (the accelerometer icons contain visual cues located above the waveform




R91-6080-039

**Figure 35. UUT computer time line for scenario no. 3.**

representation that indicate what wave shape the icon represents). The BIT time line editor controls when BIT faults are to be reported by the LRU BIT computer to the Smart BIT computer. Scenario no. 3 has 10 clusters of BIT fault reports in combinations of single impulse-type BIT faults, two single impulses separated by one OK BIT report, and bursts of consecutive BIT faults.The BIT faults occur at approximately the 13, 16, 18, 28, 31, 35, 36, 36.5, 39, and 46.5 second points in the scenario. The UUT computer's BIT time line editor also controls the length of the scenario. The total time for the third scenario is therefore about 59.5 seconds. Hence, the final temperature at the end of the scenario, as controlled by the UUT computer BIT time line, is about at the –20°C level.

The BIT and parametric information that describes a SCADC LRU for scenario no. 3 is located in the SCEN3.DAT file (see Table 5 for the contents of the SCEN3.DAT file). The RT No. and RT subaddress as well as the format and resolution of the MIL-STD-1553B message words presented in the SCEN3.DAT scenario no. 3 file were taken from the SCADC interface control document (GASD ICD-SCADC-1553). Scenario no. 3 illustrates the "Incorrect Encoded Altitude" FMECA fault scenario in which the baroset altitude word reflects a fault value and the SCADC BIT word indicates the faulted area (Baroset Corrected Altitude is invalid). All data in the SCEN3.DAT are shown in decimal format. When scenario no. 3 begins, the LRU BIT computer reports the information in the initial value column to the Smart BIT computer. As the scenario progresses (see Table 5), the corresponding parametric values in the increment value column are added periodically to the parametric values in the initial value column until the

**Table 5. Scenario No. 3 (SCEN3. DAT File)**

; SCENARIO NAME: INCORRECT ENCODED ALTITUDE

;**********

; NEW MESSAGE:

; COMMAND IS SCADC MAIN OUTPUT MESSAGE:

; RT # RT SUBADDRESS
28 16

; MODE CODE # MODE CODE DATA
99 99

| ; WORD # | DESCRIPTION | UNITS | DATA TYPE | INITIAL VALUE | INCREMENT VALUE | FAULT VALUE | FINAL VALUE | RESOLUTION | LSB BIT # |
|---|---|---|---|---|---|---|---|---|---|
| 1 | BIT_WD | NONE | BOOLEAN | 65280.0 | 0.0 | 48896.0 | 65280.0 | 1.0 | 16.0 |
| 2 | PRES_ALT | FT | SIGNED | 14000.0 | 10.0 | 14000.0 | 16000.0 | 2.5 | 16.0 |
| 3 | BARO_ALT | FT | SIGNED | 14000.0 | 10.0 | 0.0 | 16000.0 | 2.5 | 16.0 |
| 4 | TRUE_AIRSPEED | KNOTS | SIGNED | 439.0 | 2.0 | 439.0 | 507.0 | 0.125 | 15.0 |

MR91-6080-040

parameter reaches the value specified in the final value column. This simulates the dynamic conditions that would be present if the SCADC were "flying" in the aircraft. For instance, the true airspeed SCADC parameter increments from 439 knots initially to a final value of 507 knots, with an increment of 2 knots. The baroset and pressure altitudes initially start at 14,000 ft and gradually increment to 16,000 ft in 10-ft increments. The fault value column specifies the parameter value when a fault occurs. Each parameter that has a different initial value and fault value is then set to the fault value data whenever a BIT fault occurs. For instance, when the BIT word (word no. 1) is reporting no errors, its decimal value that is sent by the LRU BIT computer to the Smart BIT computer is specified in the initial value column (65280 decimal which is FF00 in hexadecimal). When an BIT fault report occurs, the LRU BIT computer reports a decimal value of 48896 (taken from the fault value column of tables) to the Smart BIT computer.

The TSMD computer reports the LRU power-on event at the beginning of the scenario in the event buffer data window (see bottom window of the TSMD computer display for scenario no. 3, Figure 36). During the course of the scenario, the short-term data window continually displays the changes in vibration peak versus time. These vibration peak data are output by the UUT computer according the scenario no. 3 vibration peak profile (see the UUT computer accelerometer time line, Figure 35). As the vibration peak passes through a threshold (shown as horizontal dotted lines in the TSMD short-term data window in Figure 36) a vibration peak event, its bin number, and the approximate time that the vibration peak event crossed the bin threshold is displayed in the event buffer data window (see the bottom window of the TSMD computer display for scenario no. 3, Figure 36). The bottom threshold of the TSMD short term data window is referred to as bin 0 and encompasses the 0 to approximately 2 g vibration peak range. The next bin is referred to as bin 1, and it represents the vibration peak range of 3 to about 6 g. The range for bin 2 is 7 to 10 g, bin 3 is 11 to 15 g, bin 4 is 16 to about 18 g, and bin 5 is approximately 19 to 25 g. The relative time (as a percentage of total scenario time) that the vibration peak remains in a particular threshold area (bin) is shown in the life stress data window (see upper right window of the TSMD computer display for scenario no. 3, Figure 36). The life stress data window indicates that the vibration peak stayed in bin 0 (bin 0 vibration peak range: 0 to approximately 2 g) approximately 41% of the time during scenario no. 3. This agrees with the UUT computer accelerometer time line shown in Figure 35 because at least 27% of the accelerometer data are level flight data (which is in the bin 0 region), with the remainder being zero accelerometer data during times in the scenario that no accelerometer waveforms are output. The life stress data window also indicates that the vibration peak stayed in bin 1 (bin 1 vibration

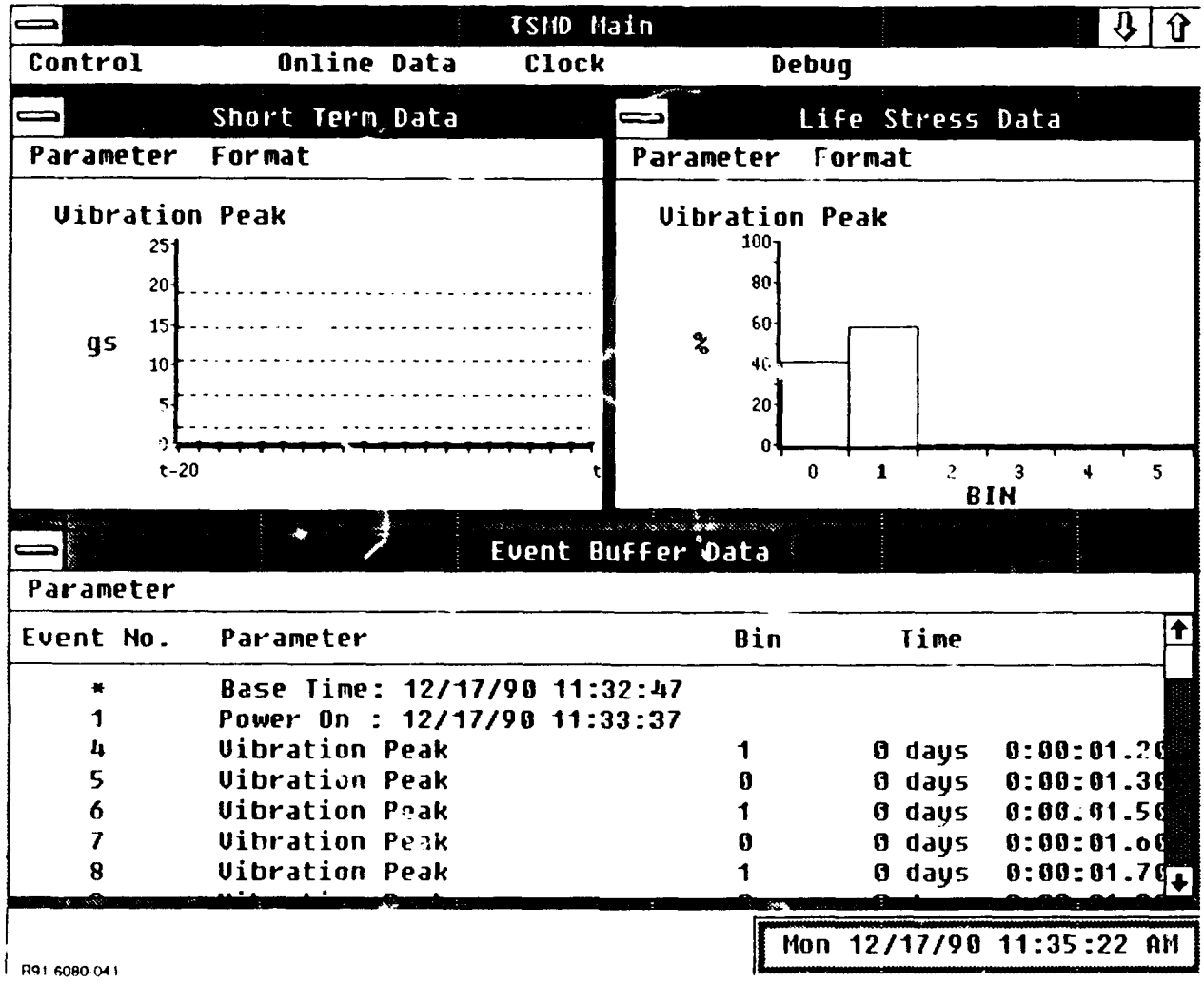
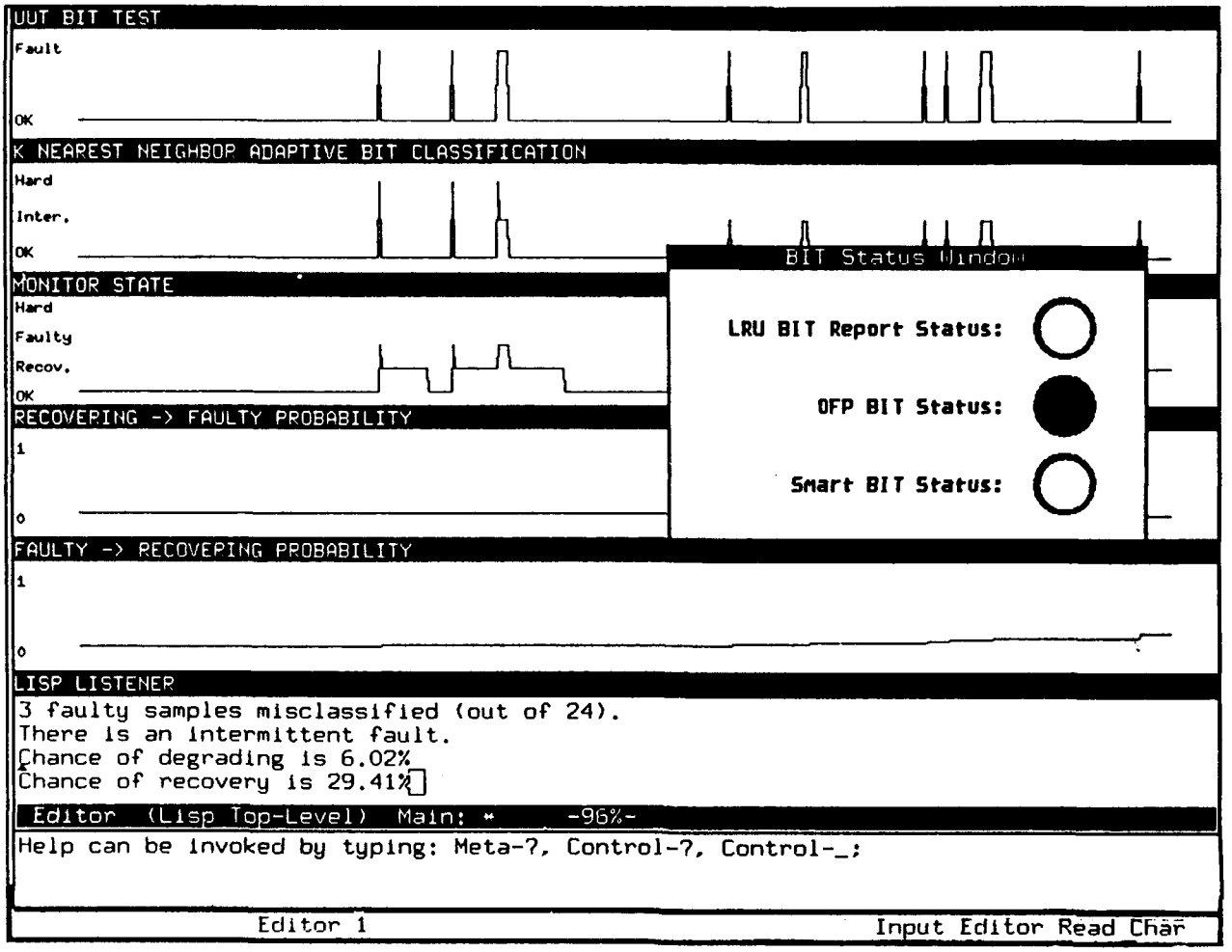



Figure 36. TSMD computer display for scenario no. 3.

peak range: 3 to approximately 6) approximately 59% of the time during scenario no. 3. This agrees with the UUT computer accelerometer time line shown in Figure 35 because most of the accelerometer waveforms in scenario no. 3 are either side slip or max g turn (with vibration peak values of about 4 g).

The Smart BIT computer UUT BIT Test window for scenario no. 3 displays the simulated SCADC LRU BIT report data for scenario no. 3 (see the Smart BIT computer display for scenario no. 3, Figure 37). It is similar to the UUT computer BIT time line editor for scenario no. 3 (see the UUT computer time line for scenario no. 3, Figure 35). There are differences between the UUT BIT test window data on the Smart BIT display and the UUT computer BIT time line data for scenario no. 3. This is due to the fact that the Smart BIT/TSMD Integration




R91-6080-042

**Figure 37. Smart BIT computer display for scenario no. 3.**

System test bed contains four asynchronously running computers with different latency times to process messages. For example, some BIT faults that the UUT computer commanded the LRU BIT computer to report might be occasionally missed by the Smart BIT computer. This is precisely why the UUT computer BIT time line and the Smart BIT UUT BIT test window data are not in total agreement. Running the same scenario again may allow the Smart BIT computer to capture all of the BIT fault reports. This turns out to be more reflective of actual avionic systems than having all computers synchronously locked to each other.

The Smart BIT K-Nearest Neighbor Adaptive BIT classification window reflects the output of the K-Nearest Neighbor algorithm. While scenario no. 3 is running, this window is updated with the latest classification of data (LRU and TSMD data). At first, the output classification incorrectly classifies the LRU and TSMD data during BIT fault reports. This is due to the fact

that the K-Nearest Neighbor algorithm must attain the minimum number of neighbors (if K = 5, then 4 other neighbors would be required) of similar values to properly classify the BIT fault reports. Once enough of the BIT and TSMD environmental data has been read in to allow proper classification, the K-Nearest Neighbor Adaptive BIT classification window mimics the data in the UUT BIT test window (see the Smart BIT computer display for scenario no. 3, Figure 37).

The Temporal Monitor analyzes the same BIT data as the K-Nearest Neighbor algorithm, but attempts to classify the input data according to their past history. When an BIT fault report is encountered, the Temporal Monitor changes its state to Intermittent (Faulty). It remains in this state for each BIT fault report. Once the BIT report returns to an OK condition, the Temporal Monitor changes its state to Recov. (Recovering). It remains in this state for successive OK BIT reports until the transition probabilities indicate that it is time to return to the OK state. The Recovering to Faulty, and Faulty to Recovering probabilities are shown in the two line graph windows below the Temporal Monitor state window (see the Smart BIT computer display for scenario no. 3, Figure 37). The Smart BIT Temporal Monitor data for scenario no. 3 indicates that the more BIT fault transitions encountered, the larger the changes in the transition probabilities.

The BIT status window contains three "lamp" indicators that change color (green for OK and red for Fault) according to the input data (see the BIT status window in the Smart BIT computer display for scenario no. 3, Figure 37). Subsection 5.1 provides a complete description of this window's functionality.

The LISP LISTENER window prints out the Smart BIT assessment for the scenario no. 3 data. It states correctly that an intermittent fault has occurred, the chance that the LRU may become intermittent is about 6.02%, and the chance for the LRU to remain OK is about 29.41%. As explained earlier in this subsection, these probabilities may be different if scenario no. 3 is run again, because the Smart BIT/TSMD Integration System test bed contains four asynchronously running computers with different latency times to process messages.

# 6 – APPROACHES FOR INSTALLING ACTUAL UUT HARDWARE

The assessment and evaluation of Smart BIT and TSMD techniques requires a means of stimulating those techniques with interesting and repeatable data. Interesting data are data which reflect the anticipated behavior of BIT reports and environmental sensors during conditions which may cause BIT false alarm reports. Repeatable data are necessary in order to vary or modify the techniques and determine the behavioral difference when subjected to the identical stimulating data set. Through the use of interesting and repeatable data, the Smart BIT and TSMD techniques can be rigorously and methodically evaluated for their effectiveness and capacities.

The Smart BIT/TSMD Integration test bed uses two different computers to provide interesting and repeatable data: the UUT computer, which simulates the TSMD sensor signals and controls the simulated BIT behavior; and the LRU BIT computer, which simulates an LRU's BIT behavior and provides a location for housing the TSMD signal conditioning circuitry. This system, shown in Figure 38, provides the test bed operator great flexibility in generating and running scenarios. Very complicated temperature and vibration profiles can be generated using

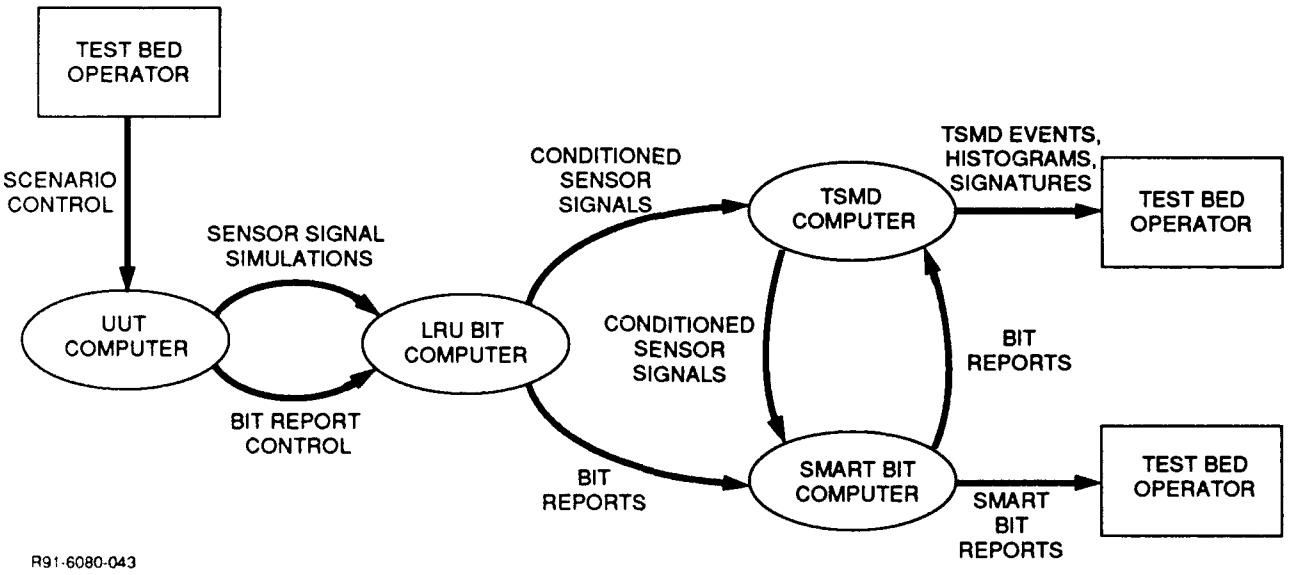



**Figure 38. UUT and LRU BIT computers provide interesting and repeatable data.**

graphical tools on the UUT computer. BIT scenario data can be created and modified easily to stimulate a variety of types of BIT reports which may represent no faults, hard faults, and different classes of false alarms. The correlation of temperature, vibration, and BIT behavior are determined by the operator according to the location of the events on the timeline. These tools have been designed to provide the operator with the ability to create and present realistic, interesting, and repeatable data to the Smart BIT and TSMD techniques.

## 6.1 GENERAL CONSIDERATIONS AND SELECTION CRITERIA

Several approaches are possible for using actual LRU hardware. However, when examining the various approaches, it must be noted that all LRU hardware is unique and would have to be examined on an individual basis prior to integration into the test bed. Figure 39 illustrates the interfaces that must be examined when considering the integration of LRU hardware into the test bed. Complete engineering data would be required to interface and/or modify the circuit board. The following information provides some cursory guidelines which may prove useful in the selection and implementation of actual LRU hardware.

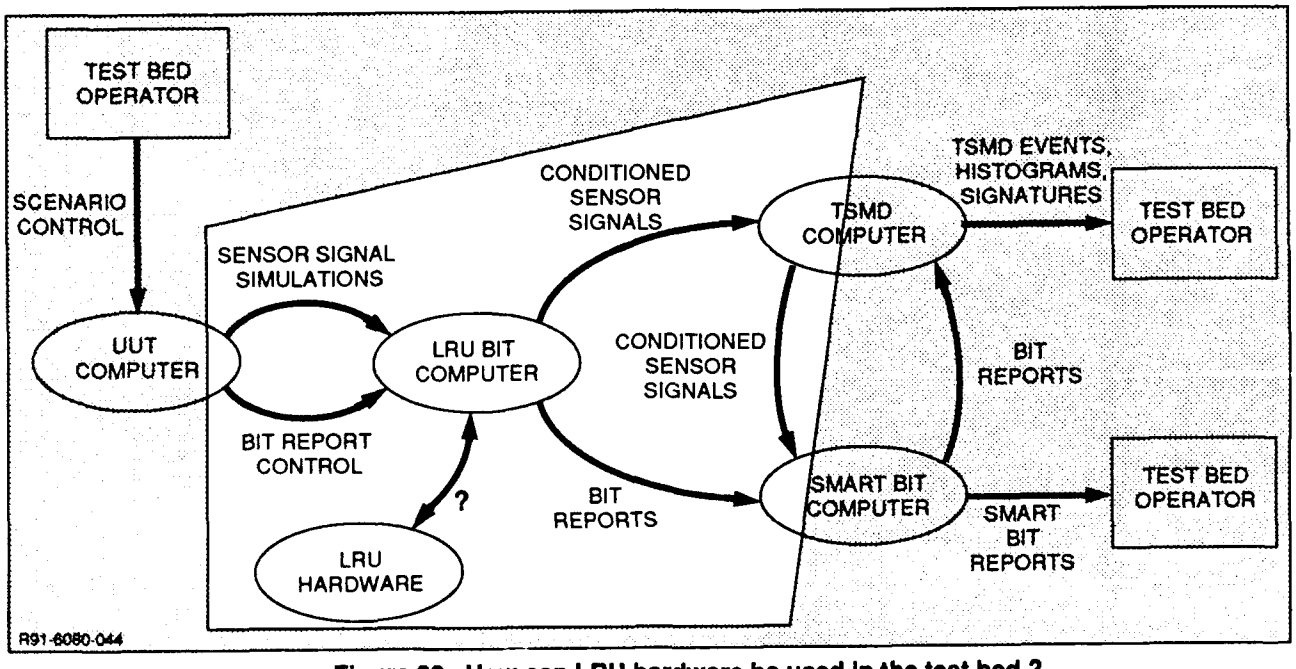



**Figure 39. How can LRU hardware be used in the test bed ?**

In order to be functionally compatible with the Smart BIT/TSMD test bed, the following criteria must be satisfied for any configuration of actual and/or simulated LRU UUT:

- The test vehicle (actual and/or simulated UUT) must report BIT results over the MIL-STD-

1553B bus interface. This is necessary to interface with the Smart BIT computer

- The test vehicle must be capable of mounting the TSMD accelerometer and temperature sensors in locations which will provide meaningful data. These sensors may be located remotely from the LRU BIT computer and connected by a 5-ft cable
- Physical stress must be applied only to the UUT, where the sensors are mounted. Physical stress must not be applied to those parts of the test bed system which are Commercial Off-The-Shelf (COTS) hardware, which have not been designed to withstand the applied physical stress. In particular, the LRU BIT computer must not be subjected to temperature or vibration extremes. Consult the COTS user's manuals for these specifications.

Two general implementation categories are possible:

- Install actual LRU hardware in the LRU BIT computer
- Adapt actual LRU hardware to interface with the test bed system.

## 6.2 INSTALLING LRU HARDWARE IN THE LRU BIT COMPUTER

Installing LRU hardware in the LRU BIT computer, although possible, may not be desirable. The LRU BIT computer is a commercial system, containing a floppy disk, a hard disk, and a cooling fan which may prematurely fail if exposed to vigorous temperature and vibration stimulus. Consult the manufacturer's manuals carefully before exposing the commercial equipment to temperature and/or vibration extremes.

The LRU BIT computer is VME-based commercial chassis containing a 68020 microprocessor, MIL-STD-1553B bus interface, and a TSMD sensor board. The system uses the OS-9 operating system and has been programmed to simulate the MIL-STD-1553B bus behavior of an LRU called a SCADC.

The SCADC simulation using a VME bus computer is very effective in providing a stimulus for both the TSMD and Smart BIT computers. The MIL-STD-1553B bus behavior of the SCADC is adequately duplicated to respond to the Smart BIT computer's queries for BIT reports; it also provides some dynamic data simulation to make the bus transfers realistic.

This approach would be to install a board which is form-fit compatible with a VME card rack. Radstone, the manufacturer of the selected hardware system, is one of several manufacturers which have delivered military qualified VME-style boards. However, caution

must be exercised when installing multiple VME bus masters, or peripherals which may occupy the same memory map addresses.

The installation of actual LRU circuit cards in the LRU BIT computer would require careful selection to ensure form, fit, and function compatibility. The LRU BIT computer uses a VME bus backplane dedicated on the P1 connector. The backplane also provides a connector for the P2 connector, but only the VME-specified power and ground pins are dedicated.

In addition to the physical and electrical power interfaces, the control and data interfaces would also have to be accommodated in order for the LRU hardware to function as it would if installed in its own LRU chassis in an operational system.

Software modifications would be necessary within the Smart BIT computer to modify the MIL-STD-1553B bus interface protocol to match the LRU selected. Additional data structures may also require modification.

The TSMD computer software would not need to be modified to integrate an actual LRU. Only the parameters used for thresholding may require alteration to accommodate the typical, quiescent temperature and vibration values experienced by the LRU. These parameters are readily altered via the TSMD software's user interface.

The UUT computer should be disconnected when integrating external LRU hardware into the test bed system. When disconnected, the TSMD sensor board input defaults to the physical sensor input, which is the desired option. The BIT simulation control feature of the UUT computer would not be used since the actual LRU hardware would be providing the BIT reports to the Smart BIT computer. If the UUT computer were to remain connected, the test bed operator would be required to make certain that the TSMD signal input selection was set to select the physical sensors.

Due to the complexities of installing actual LRU hardware in the LRU BIT computer, this approach is not recommended.

## 6.3 INTERFACING EXTERNAL LRU HARDWARE TO THE TEST BED SYSTEM

This approach, which takes advantage of the remotely located sensors, would be to mount the

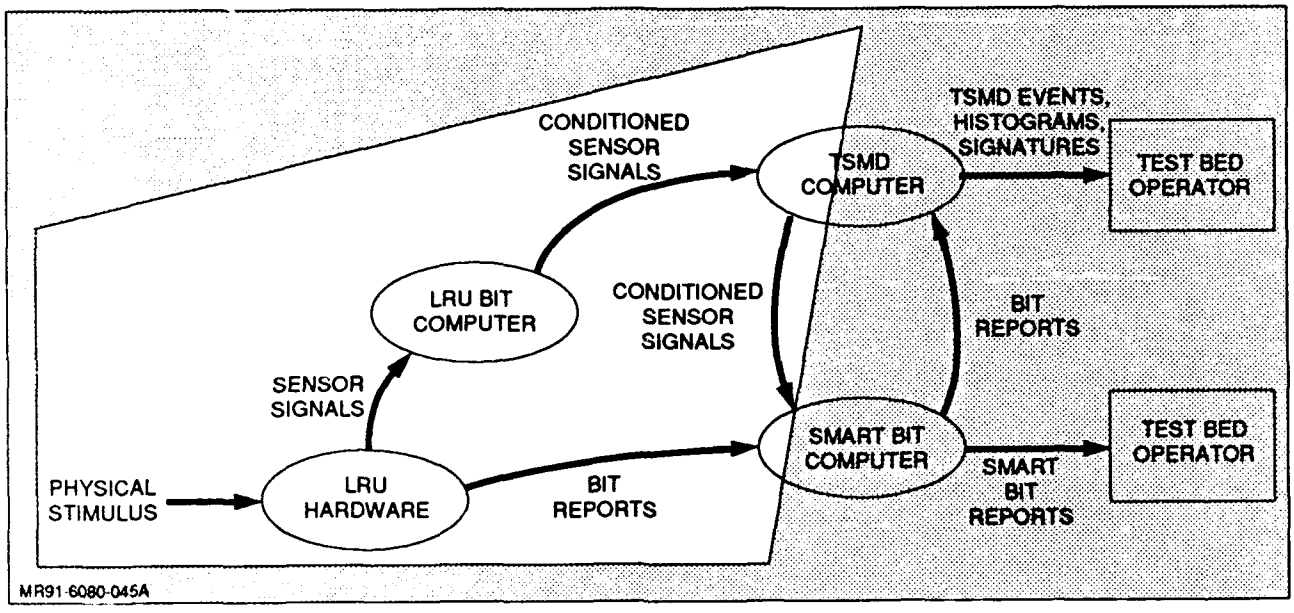



**Figure 40. Interface considerations when using LRU hardware.**

sensors inside of an actual LRU (see Figure 40). The physical stimulus could be provided by a vibration fixture placed inside of an environmental chamber to perform temperature cycling. This is similar to the facilities used to perform Environmental Stress Screening (ESS). Ideally, the Smart BIT/TSMD integration test bed could be relocated adjacent to the equipment used to perform ESS on the LRU, and the MIL-STD-1553B bus and sensor assembly cables passed through the chamber's airlock to reach the LRU.

A potential difficulty in using actual LRU hardware may be the inability to readily produce interesting data. While the physical stimulus could be provided by the ESS facilities, BIT reports may not prove to be interesting (particularly if using a "healthy" LRU). However, in line with the ESS general concept, it may be possible to locate LRU hardware which has failed the ESS testing and to use the failed unit to exercise the Smart BIT and TSMD techniques. This would not only provide actual LRU hardware for use with the test bed, but would provide an LRU with a verified environmentally-related fault condition.

By locating existing ESS facilities, the problems associated with developing a custom test setup would be avoided. These problems include mechanical test fixtures to support the LRU during vibration tests which would also have to incorporate any special cooling plenums and interfaces; electrical power supply requirements for operating the LRU; any control and data interfaces required to permit normal operation, in addition to the required MIL-STD-1553B bus interface to the Smart BIT computer; and mounting of the temperature and accelerometer sensors.

Software modifications would be necessary within the Smart BIT computer to modify the MIL-STD-1553B bus interface protocol to match the LRU selected. Additional data structures may also require modification.

The TSMD computer software would not need to be modified to integrate an actual LRU. Only the parameters used for thresholding may require alteration to accommodate the typical, quiescent temperature and vibration values experienced by the LRU. These parameters are readily altered via the TSMD software's user interface.

The UUT computer should be disconnected when integrating external LRU hardware into the test bed system. When disconnected, the TSMD sensor board input defaults to the physical sensor input, which is the desired option. The BIT simulation control feature of the UUT computer would not be used since the actual LRU hardware would be providing the BIT reports to the Smart BIT computer. If the UUT computer were to remain connected, the test bed operator would be required to make certain that the TSMD signal input selection was set to select the physical sensors.

This is the recommended approach to integrating actual LRU hardware into the test bed system.

# 7 – TEST BED CHALLENGES

The Grumman/Westinghouse team encountered many challenges during the Smart BIT/ TSMD Integration contract. These challenges included hardware and software design of four asynchronous computers that had to communicate reliably while accomplishing both their individual and integration functions in real time. This report provides the details for the hardware and software design of the Test Bed, plus the Integration factors necessary for its implementation (Sections 2, 3, and 4).

Unusual difficulties with vendor software (late delivery, immature releases, and variations of software behavior) added to the challenge. These are technical details typical of many such programs.

The overriding concerns during the course of this effort were:

- Staying as true to the real-world scenario as possible. This involved:
  - Data simulation – The use of actual aircraft environmental data recorded in flight (Subsections 2.1.1 – 2.1.9)
  - Realistic UUT scenario generation for stimulating the TSMD sensor card and controlling the simulation of BIT behavior in the LRU BIT computer (Subsection 3.1)
  - TSMD fidelity – Provide the Test Bed with the same capability as a fielded TSMD in an avionics environment (Subsection 3.3)
  - LRU simulation – Simulate a modern avionic system used in a wide variety of Air Force aircraft (Subsection 2.4)
  - Sensor signal transmission – Superimpose actual environmental transducer signals with computer generated simulations (Subsection 3.2.1.2)
  - Smart BIT analysis – Analyze the TSMD and LRU BIT data using proven Smart BIT techniques (Subsection 3.4)
  - The combination of the above factors integrated and running in real time as in an actual avionic vehicle. This meant resolving and integrating the real-time reasoning process of Smart BIT with the off-line stress data of TSMD (Section 4)
- Provide a Test Bed that was capable of simulating not just a single modern avionic system, but any LRU, with 1553 capability (Subsections 3.2.1 and 3.2.1.1)

- Provide a Test Bed that is self contained; capable of software updates with a minimum of outside dependence. *It should also be flexible and capable of supporting the assessment of* a variety of Smart BIT and TSMD concepts (Section 3, software subsections)
- The Test Bed should be capable of use as a demonstration tool (Section 4)
- Demonstrate and assess the integration of the two technologies (TSMD and Smart BIT) and identify their effectiveness, accomplished during the final technical review at Rome Laboratory.

The interfaces used in this system include RS-232C (between the UUT and LRU BIT computers), IEEE-488 (between the TSMD computer and the Smart BIT computer), MIL-STD-1553B (between the LRU BIT computer and the Smart BIT computer), and analog input and output (between the UUT computer and the sensor card residing in the LRU BIT computer chassis; and between the sensor card and the TSMD computer). Consistent message formats and software protocols for the above bus traffic were required to maintain data integrity and reliability.

The software challenges were the most demanding:

- The UUT computer software is written in the "C" programming language, with the Microsoft Windows multitasking environment. The UUT's Interface Driver software is a mix of "C" and assembly language. The assembly language was necessary to connect the software to the hardware to run in real time, to suit the requirements of our system
- The LRU BIT computer software is written in the "C" programming language and runs under the OS/9 operating system. It is designed for easy maintenance and modification to simulate any 1553 LRU
- The TSMD computer software is written in the "C" language, also under the Microsoft Windows operating environment. As with the UUT computer, the TSMD computer requires a mix of "C" and assembly language for its Interface Driver
- The Smart BIT computer is written in both LISP and "C" and is built on X11 Windows and the Unix System V version 3.2 operating system. "C" is used for its Interface Driver code, because LISP is ill-equipped to handle the Unix System V kernel driver protocol.

From the above, it is obvious that with the different languages, drivers, and operating systems required to be integrated and run in real time, some formidable software problems arose. Vendor difficulties as reported above added another dimension of complexity. In the final analysis, all problems were overcome within the program constraints.

Scenarios to demonstrate the integration of Smart BIT and TSMD technologies with environmental stress information (temperature and vibration), are discussed in Section 5. Three scenarios were chosen: temperature, vibration, and combined temperature and vibration.

Environmental studies are discussed in Section 2. As stated, it became necessary for us to derive environmental information from actual flight data for vibration, and from expert opinion for temperature. The Test Bed was designed such that it contains all the necessary information and ability to add as many scenarios and examples as desired and to change, add, or delete, as necessary.

The LRU BIT computer is a VME-based microprocessor system with a MIL-STD-1553B bus. This system was designed to provide the capability of emulating any avionic LRU that communicates on the MIL-STD-1553B interface. Thus, the Smart BIT computer has the flexibility of interfacing with literally hundreds of LRUs. The capability of installing actual UUT hardware is discussed in Section 6.

# 8 – CONCLUSIONS AND RECOMMENDATIONS

## 8.1 CONCLUSIONS

The objective of developing a Test Bed for the purpose of integrating Smart BIT and TSMD technologies to study intermittent BIT behavior as a function of stress has been accomplished. This Test Bed is the beginning of the study and not an end in itself. It is a tool to demonstrate the benefits of combining these two technologies. This involves BIT techniques, TSMD approaches, and Smart Bit analyses on a variety of Units Under Test. The Test Bed provides us with the ability to determine and assess the criteria for integrating the two technologies: that is, computer resources, peculiar timing approaches, scaling factors, interfacing requirements, and software approaches demonstrating their interaction and therefore, the necessary steps for the eventual transition from the laboratory to the field.

The following specific conclusions may be derived from the Test Bed results:

- The Test Bed can provide TSMD circuitry and techniques with the same capabilities as a fielded TSMD in an avionic environment. Conditioned sensor signals can measure, process, and compress sensor data using the algorithms of Life-stress monitoring, Over-stress event logging, and Fault event logging
- Smart BIT techniques running in real time can be successfully integrated with off-line TSMD data. The Smart BIT techniques involve analysis using Neural Networks, K-Nearest Neighbor, and Temporal Monitoring combining a Markov model finite-state machine of intermittent behavior with Bernoulli random variables. Other Smart BIT techniques may be added in the future, to build upon the above techniques for more effective analysis
- The scenarios demonstrating the integration of TSMD with Smart BIT are derived from actual aircraft environmental data recorded in flight. This demonstrates the real-world capability of the Test Bed
- The Test Bed has demonstrated its flexibility and growth potential. It allows for the study and integration of Smart BIT and TSMD while retaining their independent capability. It provides for the use of new field data as they become available and all development tools reside on the Test Bed for software modifications without external support. This includes environmental stresses that may be simulated or induced by actual sensors or a combination of both for experimentation

- The Test Bed has demonstrated the ability to allow for the realistic prototyping of any fielded MIL-STD-1553B avionic LRU. This gives it the flexibility to simulate literally hundreds of such LRUs.

In summary, the Test Bed has demonstrated its capability to perform its intended purpose.

## 8.2 RECOMMENDATIONS

The studies conducted to date have proven that the Smart BIT techniques can identify false alarms. However, they have been conducted under laboratory conditions with sophisticated computers and high-level software tools. A final application study should be conducted to reduce this sophistication and complexity to on-board avionics. This would involve selecting a suitable avionic subsystem, determining Smart BIT avionic changes (certainly software, minimum, or no hardware changes), determining Smart BIT approaches and software changes, and implementing the changes on an actual avionic subsystem.

Recommendations for increasing the scope of the Test Bed follow:

- Substitute an actual avionic LRU for the LRU BIT computer in the Test Bed. The methodology was discussed in Section 6. This would afford a first-hand approach to the eventual application to a fielded system
- Replace the present Smart BIT computer with a faster processing computer, such as a SUN workstation, or an enhanced MacIntosh with a MacIvory or micro Explorer card, to obtain faster processing speeds. Also, provide display modifications to enhance graphics/ data integration
- Conduct a study to integrate the benefits of Neural Network adaptive learning (to associate BIT fault patterns with environmental stress data) and the ability to track BIT behavior over time (Temporal Monitoring) to find the best combination of speed and classification performance.

Other investigations include:

- A study using TSMD historical data with Neural Networks. This would involve obtaining long-term historical trends of stress levels via the TSMD and analyzing these trends automatically via the Neural Networks
- The potential for using hardware Neural Network Integrated Circuits for implementing selected Smart BIT paradigms. This would free computer processing requirements and speed the analysis of Neural Network processing, many times

- Develop a Smart BIT card for advanced avionics applications, eventually leading to a Smart BIT chip. This could then be inserted into avionics units, as required
- Develop an avionic Smart BIT/TSMD specification for incorporating this combined technology into advanced avionics. Potential users would then be guided to implement Smart BIT/TSMD techniques on an as-needed basis.

The present studies concentrated on stresses due to temperature and vibration. For completeness and comparison, studies should also be conducted with other stress factors, such as power supply glitches, shock, and electro-magnetic interference.

Smart BIT techniques detect intermittents. With TSMD, Smart BIT determines the stress factors that cause the intermittent. However, Smart BIT does not yet fault isolate (i.e., diagnose) to the component(s) failing intermittently. Intermittents are difficult to diagnose, because they do not persist long enough for diagnosis to proceed very far. Techniques such as Opportunistic BIT discussed in the Smart BIT 2 Final Report, RADC-TR-89-277, show promise of accomplishing the above diagnosis and merit a serious study.

# APPENDIX A

## CALIBRATION PROGRAM FOR IFFT AND FFT

This program uses the spectral density of White noise to calibrate the IFFT and FFT equations used to convert vibration spectral density to a time function as discussed in paragraph 2.1.3 of the final report

NOTE: Equation numbers used below are according to those defined in paragraph 2.1.1 of the Final Report

This section defines some basic parameters

$Ni := 512$ The max index in the frequency domain

$Nj := 1023$ The max index in the time domain

$N := Nj + 1$ The number of points in the time domain

$fc := 2000$ The white noise cutoff frequency

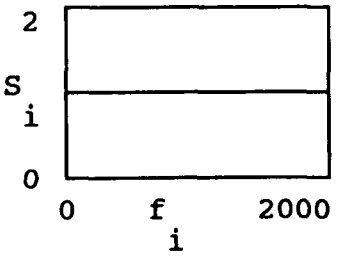
$\delta f := \frac{fc}{Ni}$ $\delta f = 3.906$

$i := 0 \,..Ni$

$S_i := 1$ By definition, the spectral density for white noise is constant over the frequency band

$f_i := i \cdot \delta f$

$BW := fc - 0$



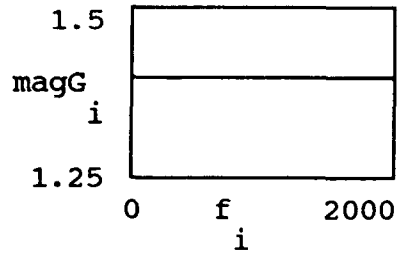

$grms := \sqrt{S_0 \cdot BW}$ The closed form equation for grms from the continuous frequency domain

$$grms1 := \sqrt{\sum_i \left[S_i \cdot \delta f\right]}$$ Equation [4]

This section derives the fast Fourier transform G(i)
from the spectral density S(i)

$$magG_i := \sqrt{\frac{S_i}{2} \cdot \delta f}$$

Equation 3
NOTE: The factor 1/2 is to account for
power at the negetive frequencies




$phsG_i := rnd(2 \cdot \pi)$

A reasonable assumption since it results in a g(j) probability function which is normal (Gaussian).

$phsG_0 := 0$

$phsG_{511} := 0$

Since g(j) is real, by definition the phase at i=0 and i=N/2-1 is 0

$$G_i := magG_i \cdot \cos\left[phsG_i\right] + magG_i \cdot \sin\left[phsG_i\right] \cdot \sqrt{-1}$$

$gmean1 := G_0$

Equation [5]

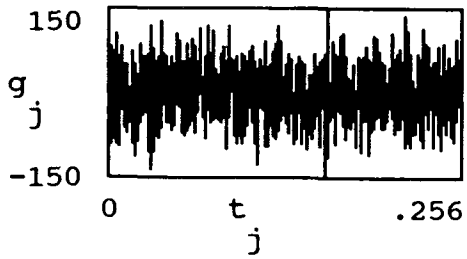
This section transforms G(i) in the frequency domain
to g(j) in the time domain

$g := \sqrt{N} \cdot ifft(G)$

Transforming G(i) to g(j) using the inverse fast Fourier transform(ifft)
NOTE: Square root of N is required because of the normalization used by the ifft function

$j := 0 \,..\, Nj$

$$TI := \frac{1}{\delta f} \qquad \delta t := \frac{TI}{Nj} \qquad t_j := j \cdot \delta t$$




min(g) = -146.289  max(g) = 149.088

gmean2 := mean(g)  grms2 := stdev(g)

This section derives the probability function of g(j) and compares it to a normal probability function

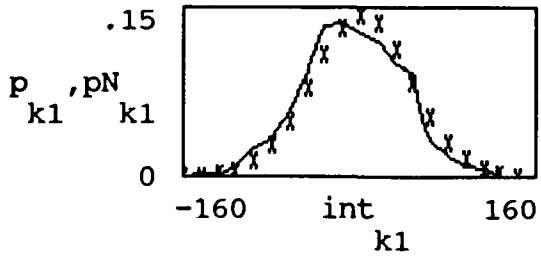
j1 := 0 ..20

int := -160 + j1·16  Defines intervals over which g(j) probability function is calculated
j1

k1 := 0 ..19

$$p := \frac{hist(int,g)}{N}$$ The probability function for g(j)

$$pN_{k1} := \left[\frac{16}{44.699\cdot\sqrt{2\cdot\pi}}\right]\cdot\exp\left[-1\cdot\frac{\left[int_{k1} - 1.398\right]^2}{2\cdot 44.699^2}\right]$$ A normal probability function



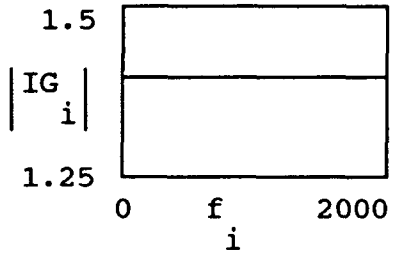

p - solid line

pN - discrete X's

This section transforms g(j) back to IG(i) and IS(i), the initial functions we started with. This shows closure accuracy.

$$IG := \frac{1}{\sqrt{N}} \cdot fft(g)$$

Transforming g(j) to G(i) using the fast Fourier transform (fft)
NOTE: Division by the square root of N is required because of the normalization used by fft

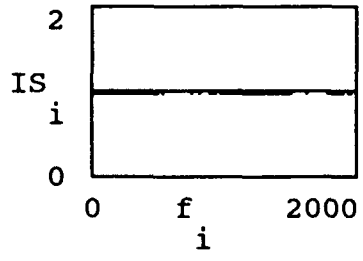



$$gmean3 := IG_0$$

Equation [5]

$$IS_i := 2 \cdot \frac{\left[\left|IG_i\right|\right]^2}{\delta f}$$

Equation [3]
NOTE: The factor 2 is to account for power at the negetive frequencies




$$grms3 := \sqrt{\sum_i IS_i \cdot \delta f}$$

Equation [4]

This section displays the final results of the grms and gmean values calculated above

| | |
|---|---|
| grms = 44.721 | (reference, closed form equation) |
| grms1 = 44.765 | (S(i) and equation [4]) |
| grms2 = 44.699 | (g(j) and equation [7]) |
| grms3 = 44.764 | (IS(i) and equation [4]) |
| gmean1 = 1.398 | (G(i) and equation [5]) |
| gmean2 = 1.398 | (g(j) and equation [6]) |
| gmean3 = 1.398 | (IG(i) and equation [5]) |

# APPENDIX B

## VIBRATION SPECTRAL DENSITIES AND TIMELINES

This appendix contains the vibration spectral densities and the respective vibration timelines for the following flight modes:

- Normal Level Flight
- Speed Brakes Out
- Level Turn Constant G
- Level Turn Max G
- Side Slip
- Weapons Drop.

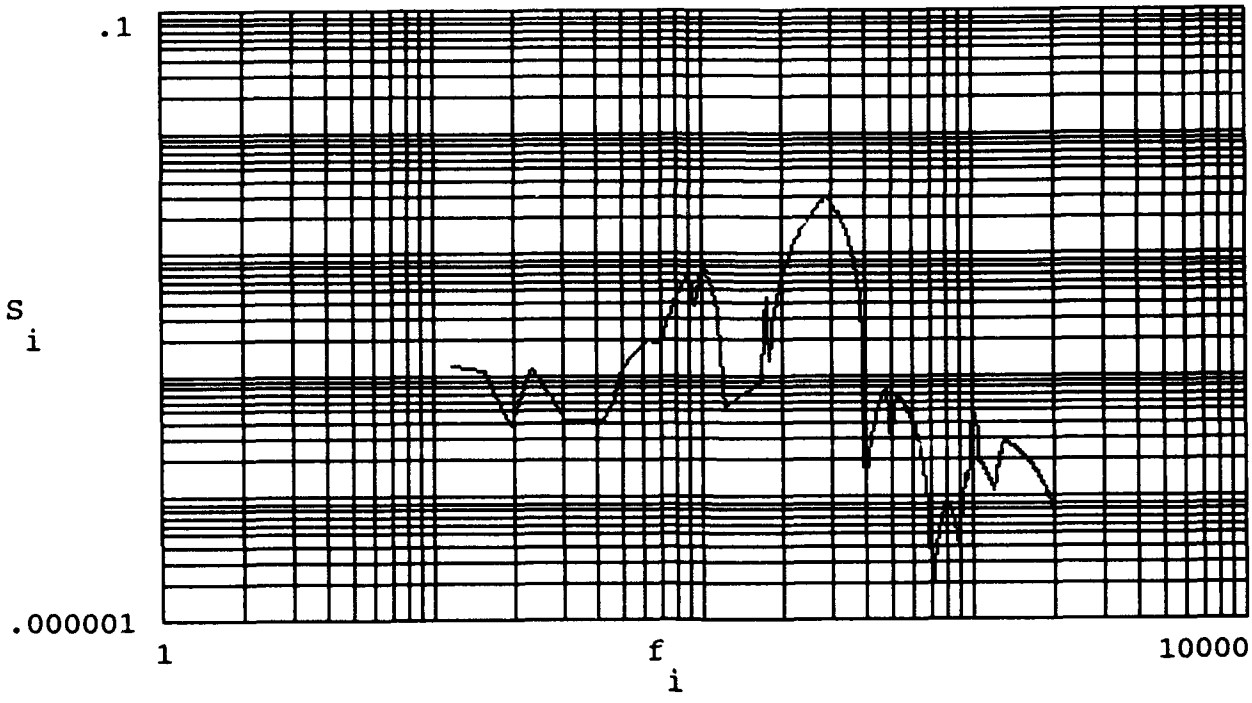
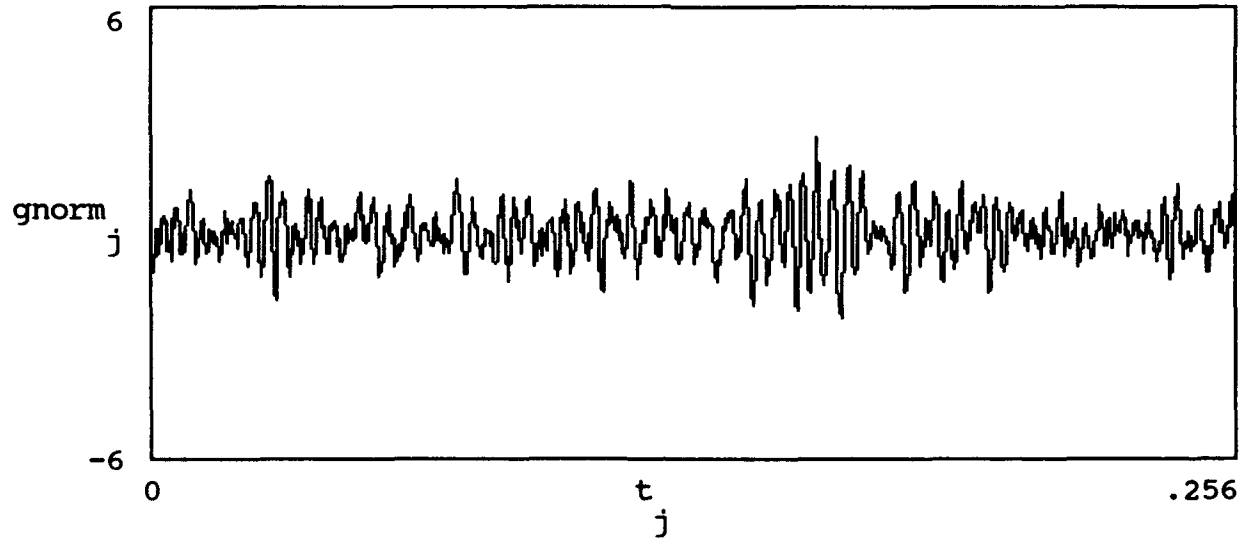



**Figure B-1. PCB vibration spectral density, normal level flight.**




min(gnorm) = -2.275 max(gnorm) = 2.535

mean(gnorm) = $4.419 \cdot 10^{-5}$ stdev(gnorm) = 0.638

**Figure B-2. PCB vibration time-line, normal level flight.**

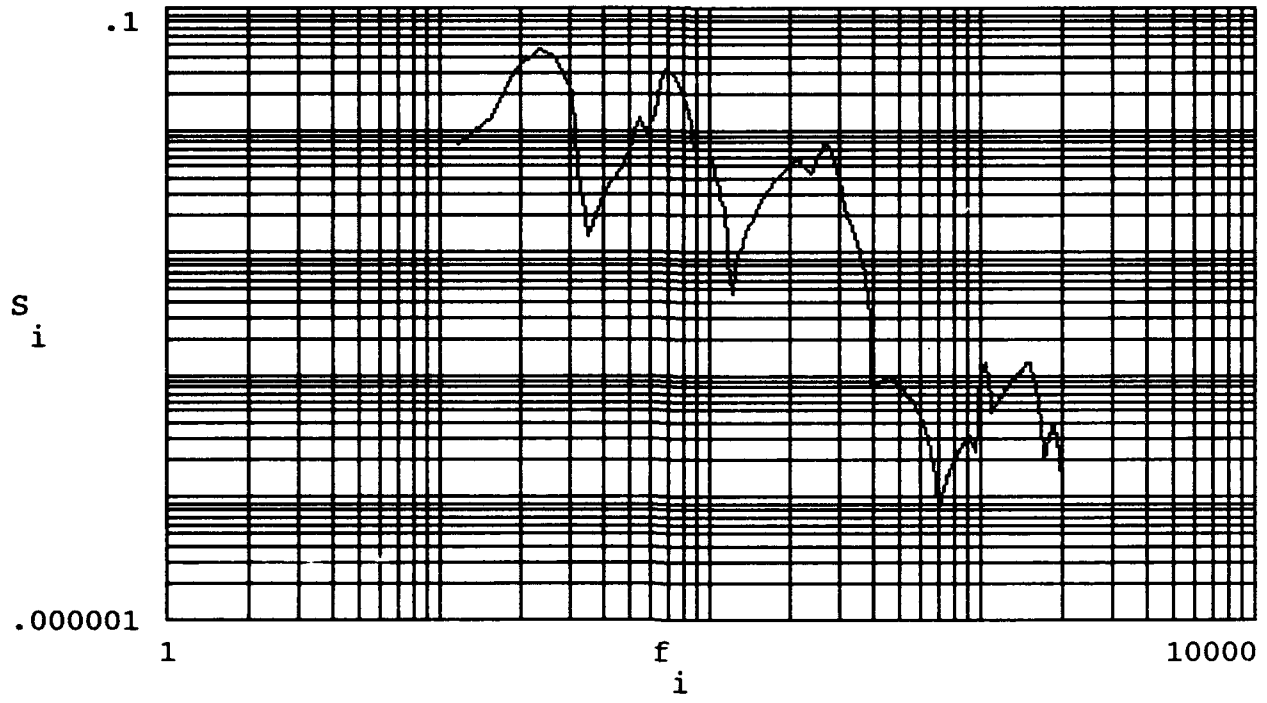
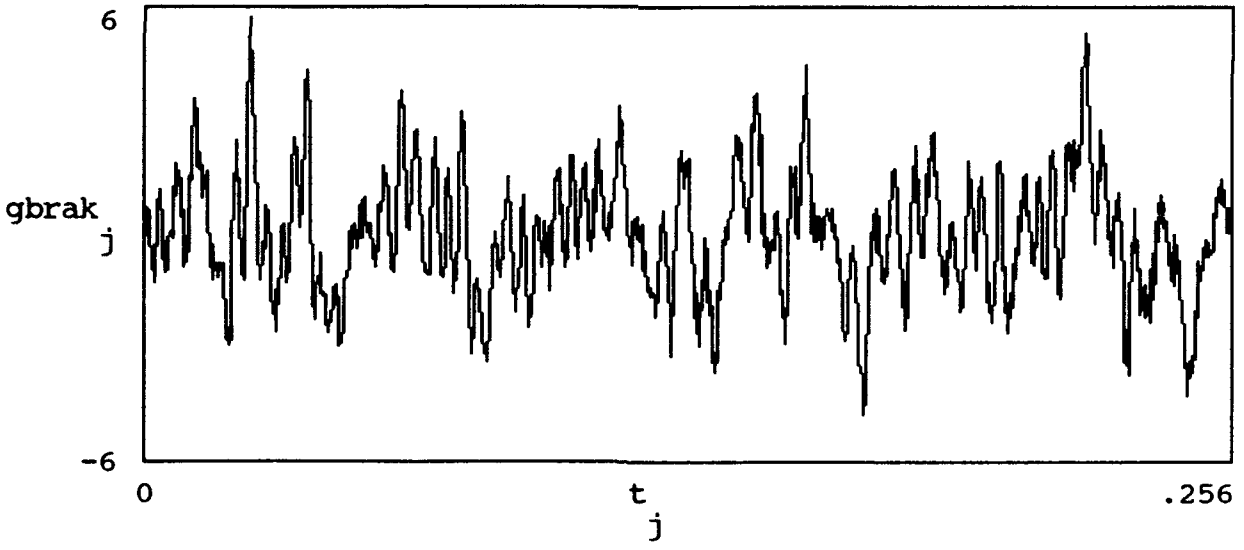



**Figure B-3. PCB vibration spectral density, speed brakes out.**




min(gbrak) = -4.762 max(gbrak) = 5.748

mean(gbrak) = $4.419 \cdot 10^{-5}$ stdev(gbrak) = 1.587

**Figure B-4. PCB vibration time-line, speed brakes out.**

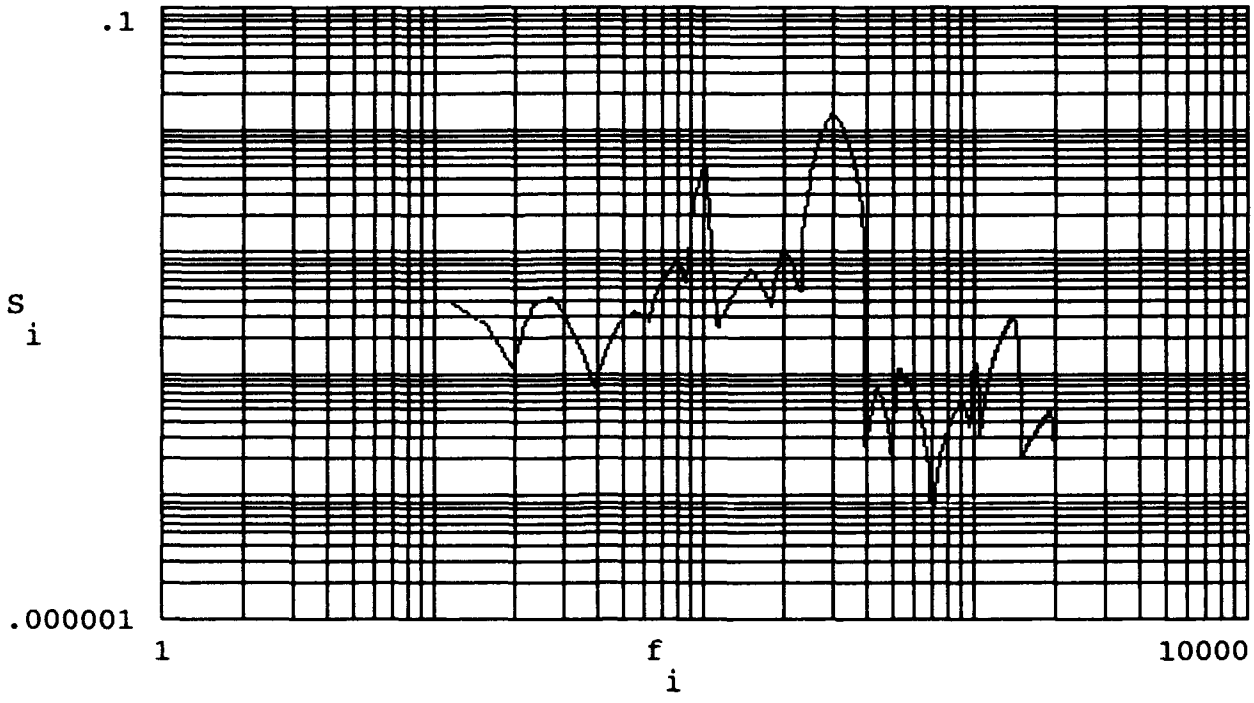
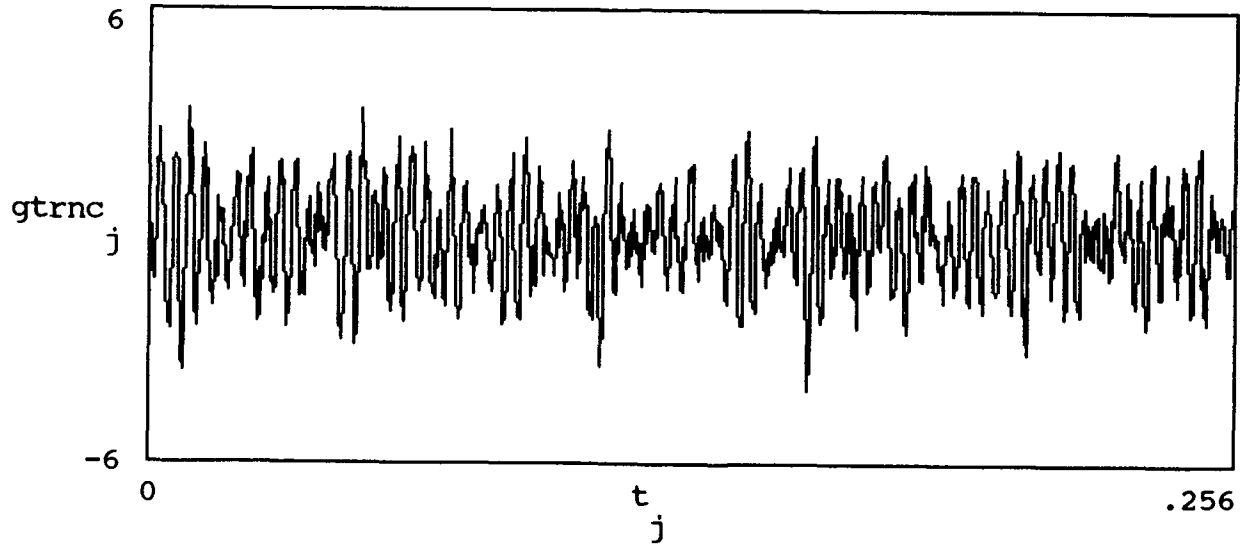



Figure B-5. PCB vibration spectral density, level turn constant G.




min(gtrnc) = -4.086 max(gtrnc) = 3.345

mean(gtrnc) = $4.419 \cdot 10^{-5}$ stdev(gtrnc) = 1.224

Figure B-6. PCB vibration time-line, level turn constant G.

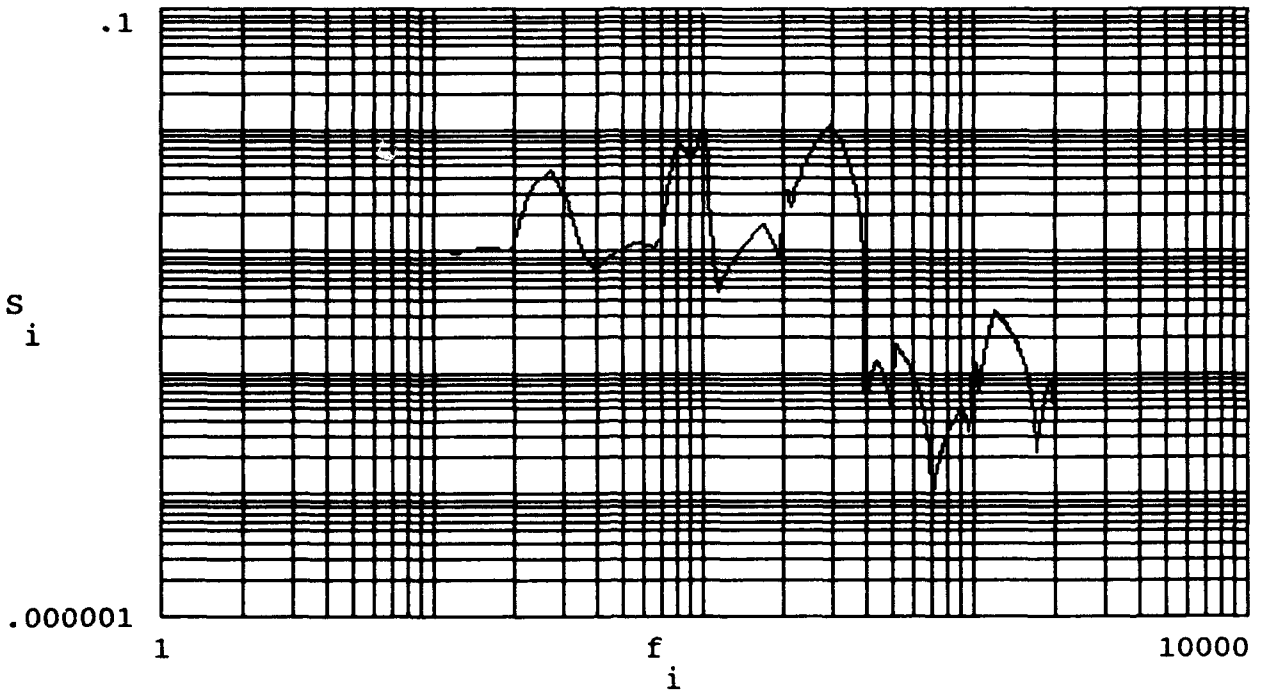
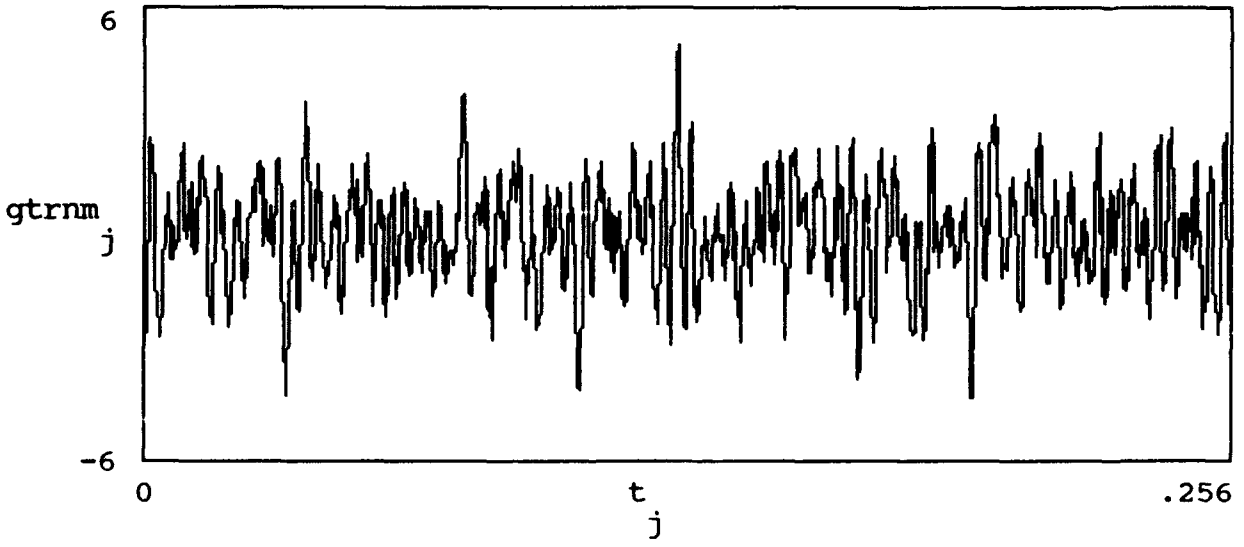



**Figure B-7. PCB vibration spectral density, level turn max G.**




min(gtrnm) = -4.331 max(gtrnm) = 5.049

mean(gtrnm) = $4.419 \cdot 10^{-5}$ stdev(gtrnm) = 1.337

**Figure B-8. PCB vibration time-line, level turn max G.**

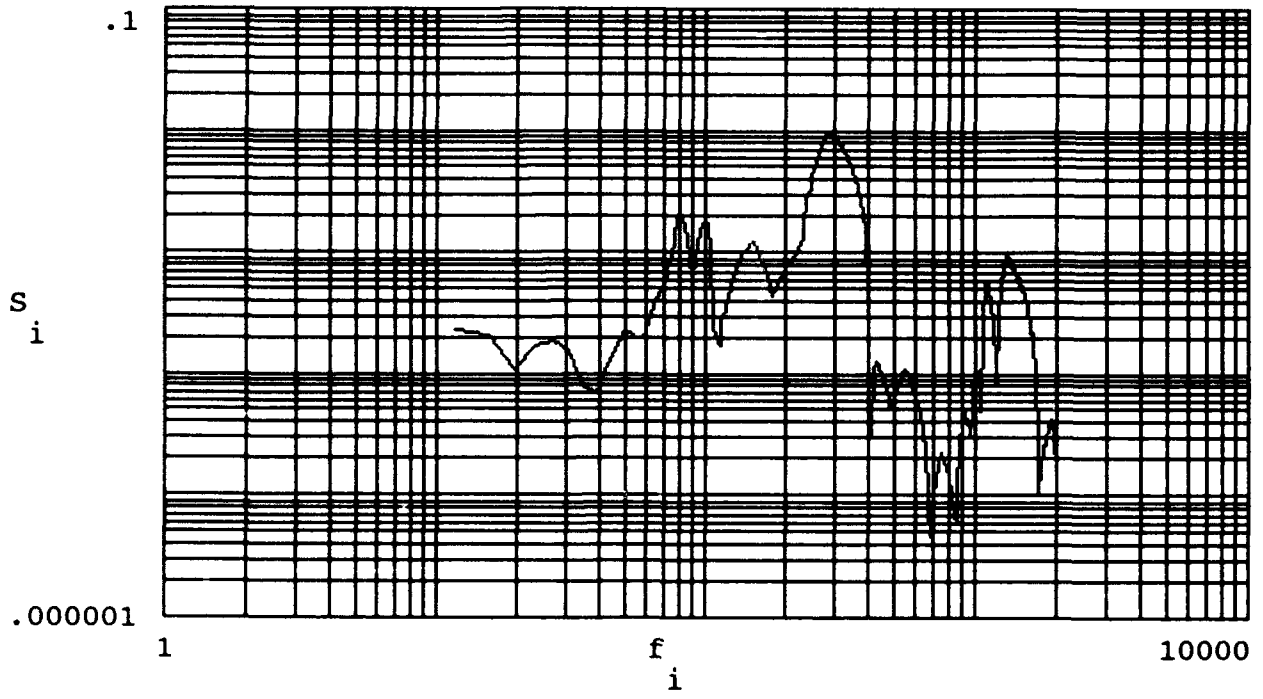
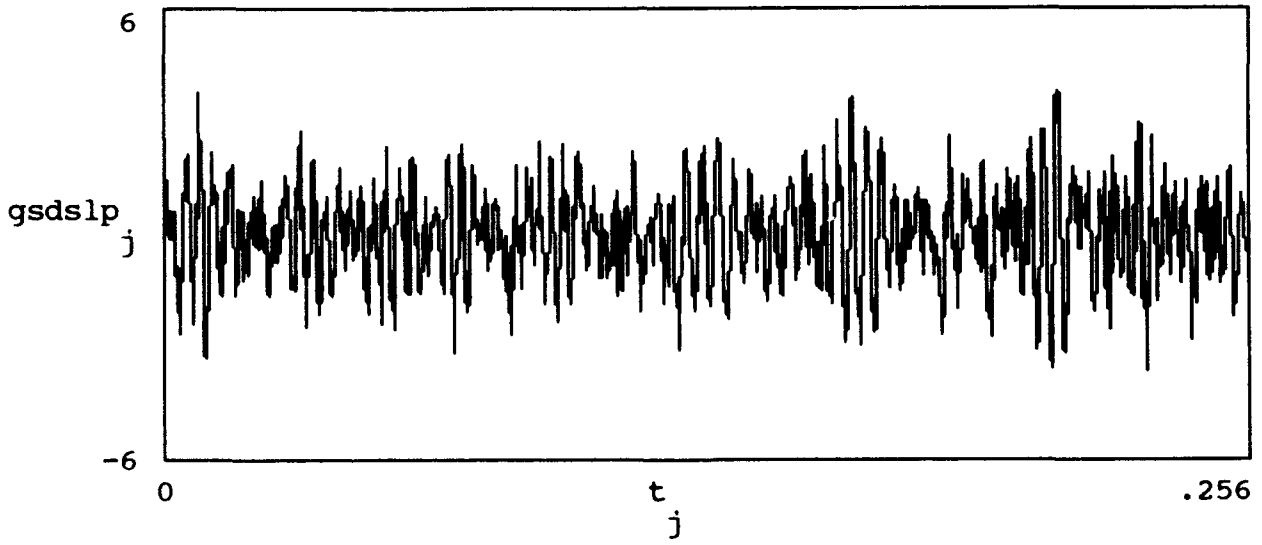



Figure B-9. PCB vibration spectral density, side slip.




min(gsdslp) = -3.645 max(gsdslp) = 3.788

mean(gsdslp) = $4.419 \cdot 10^{-5}$ stdev(gsdslp) = 1.195

Figure B-10. PCB vibration time-line, side slip.

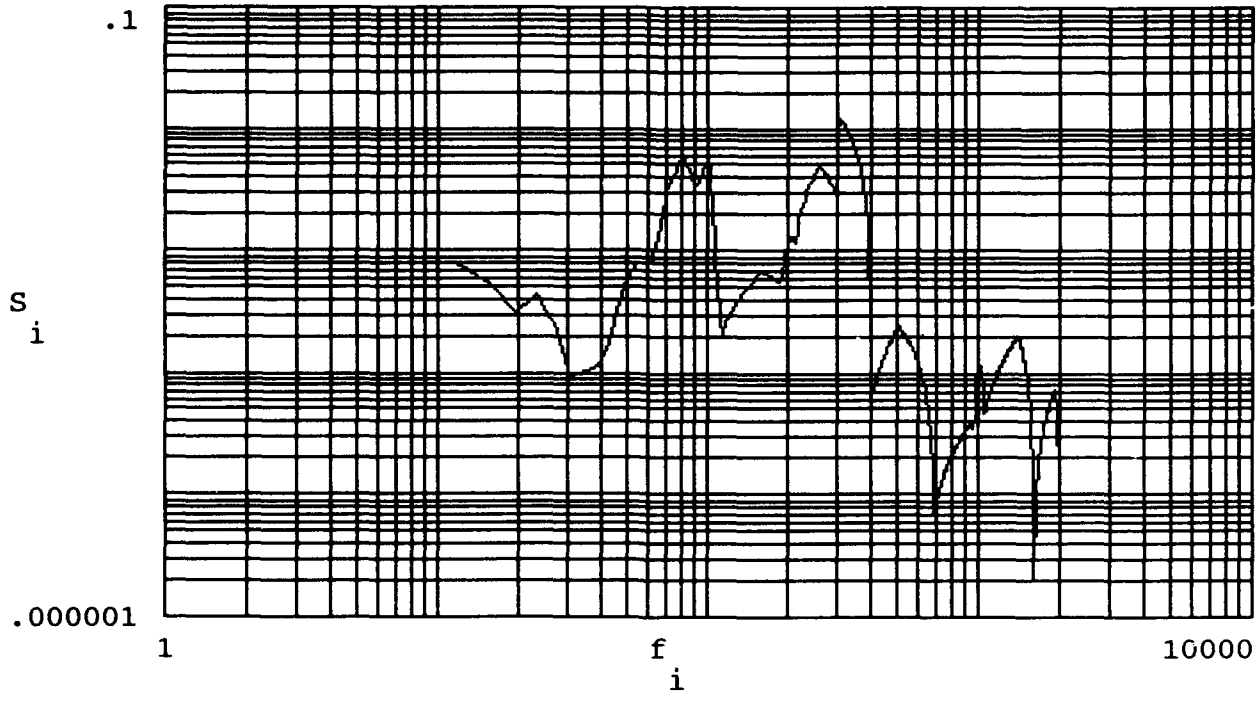
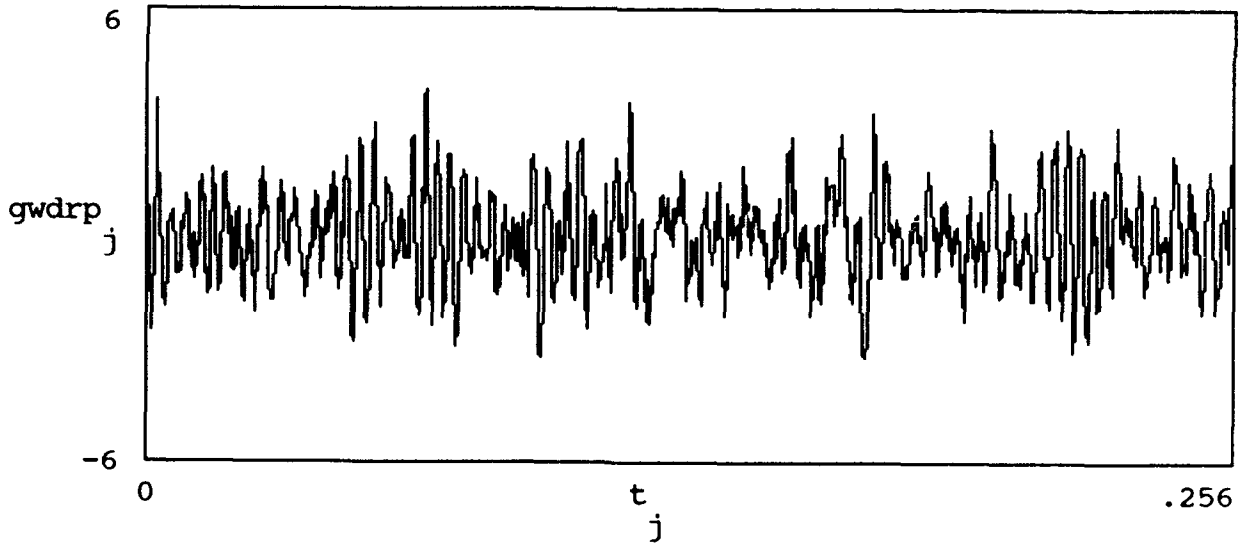



**Figure B-11. PCB vibration spectral density, weapons drop.**




min(gwdrp) = -3.221 max(gwdrp) = 3.862

mean(gwdrp) = $4.419 \cdot 10^{-5}$ stdev(gwdrp) = 1.161

**Figure B-12. PCB vibration time-line, weapons drop.**

# APPENDIX C

## LIST OF SUPPORTING DOCUMENTATION

### C.1 REFERENCES


1. Haller, K.A., Zbytniewski, J.D., Anderson, K., and Bagnall, L., "Smart BIT," RADC TR-85-148.
2. Haller, K.A., Zbytniewski, J.D., Anderson, K., "Smart Built-In Test (BIT)" IEEE Autoestcon, 1985.
3. Cooper, C., Haller, K., Zbtniewski, J., Anderson, K., and Matson, J., "Smart BIT-2" RADC-TR-89-277, December 1989.
4. Zbytniewski, J.D., Anderson, K., "Smart BIT-2: Adding Intelligence to Built-In Test," IEEE NAECON Proceedings – 1989.
5. Marchiel, C., Haller, K., Zbytniewski, J., Smart BIT for Joint Stars, Final Report, Contract No. F30602-87-C-0174.
6. Zourides, V.G., "Smart Built-In-Test (BIT) An Overview," IEEE Autotestcon, 1989.
7. Chinn, J. and Bolds, P., "Determination of the Dynamic Environment of the F/FB-111 Tail Pod Assembly," AFWAL, Shock and Vibration Bulletin, Part 3, May 1982.
8. "Vibration Sources and Relationships," Task 1: Design Definition Phase for Micro TSMD Development, ASFC, Contract No. F30602-87-D-0152, March 1989.
9. Symkowiak, E.A., and Silver, W., "A Captive Store Flight Vibration Simulation Project," Institute of Environmental Science Proccedings, 1990.
10. Symkowiak, E.A. and Gruenberger, H.S., "Empirical Determination of Damage Threshold for Leadless Chip Carriers on Printed Wiring Boards," Institute of Environmental Science Proceedings, 1986.


Additional documentation was produced and delivered to the Air Force under this contract, and includes the following:

- Hazard Analysis
- Test Plan/Procedures
- Software Test Plan
- Design Plan
- Engineering Drawings

- Interface Design Document
- Software Design Document
- Software Programmer's Manual
- Commercial Off-the-Shelf Manuals.

## C.2 BIBLIOGRAPHY

The references shown in this bibliography were compiled from conference proceedings, papers, and contract study reports used by Grumman and Westinghouse during the Smart BIT/ TSMD Integration project.


| | |
|---|---|
| GEC Pub. No. 29338 | Failure Mode, Effect and Criticality Analysis Report (FMECA) for the SCADC CPU-142/A |
| GEC Spec. No. 29039/1C | Prime Item Development Specification for the SCADC CPU-142/A |
| GEC Pub. No. 260/30231 | Computer Program Development Specification for SCADC CPU-142/A |
| GEC Pub No. 260/30252 | Computer Program Product Specification for SCADC CPU-142/A, Volumes 1 – 10 |
| GASD ICD-SCADC-1553 | Interface Control Document for the SCADC MIL-STD-1553B Signal Data Interface |
| AF T.O. IF-111A-2-16-1S-6 | Organizational Maintenance Air Data Computer Systems |

Smart BIT/TSMD Integration Statement of Work (SOW) (PR No. N-9-5507)

Interface Design Document for the Smart BIT/TSMD Integration Contract CDRL A008 (Contract No. F30602-89-C-0155)

Design Plan for the Smart BIT/TSMD Integration Contract CDRL A006 (Contract No. F30602-89-C-0155)

T.G. Dietterich, "Learning about Systems That Contain State Variables," Proc. of the National Conference on AI, American Assoc. for AI, August 1984, pp 96, State, Learning.

B. Leban, D. McDonald and D. Forster, "A Representation for Collections of Temporal Intervals," Proc. of the 5th National Conference on AI, American Assoc. for AI, August 1986, pp. 401

J. Sticklen, B. Chandraseekaran, and J.R. Josephson, "Control Issues in Classificatory Diagnosis," Proc. of the 9th International Joint Conference on AI, August 1985, pp. 300-306, Diagnosis

J.F. Allen and P.J. Hayes, "A Common-Sense Theory of Time," Proc. of the 9th International Joint Conference on AI, August 1985, pp. 528-531, Events

G.D. Blank, "A New Kind of Finite State Automation: Register Vector Grammar," Proc. of the 9th International Conference on AI, August 1985, pp. 749-755, State

D.W. Richards, "Smart BIT: A Plan For Intelligent Built-In-Test," IEEE AES Magazine, Jan. 1989, pp 26-29

C. Cooper, K. Haller, J. Zbytniewski, K. Anderson, and J. Matson, "Smart BIT-2," RADC-TR-89-277, December 1989

R.P. Lippmann, "An Introduction to Computing With Neural Nets," IEEE ASSP Magazine, April 1978, pp. 4-22

R.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing: Exploration into the Microstructure of Cognition, MIT Press, 1986

Marvin Minsky, Seymore Papert, Perceptrons: "An Introduction to Computational Geometry," MIT Press, 1969.

B. Widrow and M.E. Huff, "Adaptive Switching Circuits," IRE WESCON Conv. Record, Part 4, pp 96-104, 1960

B. Widrow and S.D. Stearns, "Adaptive Signal Processing" 1985

D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representation by Error Propagation," in R.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing, Exploration into the Microstructure of Cognition, MIT Press, 1986

J.J. Hopfield, and D.W. Tank, "Computing with Neural Circuits: A Model," Science, Vol. 233. 625-633, August 1986

D.W. Tank and J.J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," IEEE Trans. Circuits Systems CAS-33, 553-541, 1986

M. Caudill and C. Butler, "Naturally Intelligent Systems," 1990.

"Vibration Sources and Relationships." Task 1: Design Definition Phase for Micro Time Stress Measurement Device (TSMD) Development, AFSC Contract No. F30602–87–D–0152, March, 1989.

## APPENDIX D

## LIST OF ACRONYMS

| | |
|---|---|
| AAG | Aircraft Adapter Group |
| ACP | Auxiliary Communications Panel |
| AFL | Avionics Fault List |
| AI | Artificial Intelligence |
| AIS | Avionic Intermediate Shop |
| AMC | Advanced Microelectric Converter |
| AMP | Avionics Modernization Program |
| AMUX | Avionics Multiplex Bus |
| APP | Avionics Power Panel |
| ASCII | American National Standard Code For Information Interchange |
| ASP | Armament Select Panel |
| ATMS | Assumption-based TMS |
| AVTR | Airborne Video Tape Recorder |
| BC | Bus Controller (MIL-STD-1553B) |
| BIT | Built-In-Test |
| BNS | Bombing/Navigation System |
| CND | Cannot Duplicate |
| DAC | Digital to Analog Converter |
| DTS | Data Transfer System |
| FA | False Alarm |
| FF | Flip-Flop |
| FMECA | Failure Mode, Effect and Criticality Analysis |
| FSM | Finite State Machine |

| | |
|---|---|
| GFE | Government Furnished Equipment |
| GNC | General Navigation Computer |
| HF | High Frequency |
| ILS | Instrument Landing System |
| KNN | K – Nearest Neighbor |
| LRU | Line Replaceable Unit |
| MC | Mission Computer |
| MTCA | Multiple Transformer Coupler Assemblies |
| NCU | Navigation Computer Unit |
| NN | Neural Network |
| OFP | Operational Flight Program |
| PC | Personal Computer |
| PCB | Printed Circuit Board |
| RADC | Rome Air Development Center (now Rome Laboratory) |
| RAM | Random Access Memory |
| RI | Radar Indicator |
| ROM | Read Only Memory |
| RRU | Remote Readout Unit |
| RSFF | Reset Set Flip-flop |
| RT | Remote Terminal (MIL-STD-1553B) |
| RTOK | Retest O.K. |
| SCADC | Standard Central Air Data Computer |
| SRU | Shop Replaceable Unit |
| ST | Self Test |
| TSMD | Time Stress Measurement Device |

| | |
|---|---|
| UUT | Unit Under Test |
| VLSI | Very Large Scale Integrated |
| WCP | Weapons Central Panel |
| WDC | Weapons Delivery Computer |
| WDT | Watch Dog Timer |
| WSO | Weapon Systems Operator |
| XOR | Exclusive OR |

# Appendix E
# Smart Bit Techniques

A thorough analysis of the SCADC's conventional BIT was performed, resulting in the following conclusions:

- BIT tests are too rigid. Once fooled, they can be fooled again
- BIT testing, at best, uses a limited, rigid model of intermittent behavior, e.g., "M out of N" (discussed in Subsection E.3)
- No information about intermittents is provided by BIT. Systems states are "OK" or "Faulty"
- Environmental stress is not accounted for.

As a result of the AI and BIT assessment and in light of the previous Smart BIT techniques, three new Smart BIT techniques were derived that may potentially reduce the false alarm rate of avionic BIT. They are:

- Adaptive BIT - Recognizing intermittent behavior by identifying situations where intermittents have occurred previously
- Temporal Monitoring BIT - Develop a model of faulty behavior over time
- Opportunistic Diagnostic BIT - Extracting as much diagnostic information as possible out of each failed BIT test.

These approaches are interested in (1) estimating the probability that a fault is intermittent, and (2) extracting whatever diagnostic information is available about the fault. Of the three approaches, Opportunistic Diagnosis is primarily a diagnostic tool, while Adaptive BIT and Temporal Monitoring BIT are primarily useful for identifying faulty behavior while providing some information that may be useful in diagnosis.

Both adaptive BIT and Temporal Monitoring BIT attempt to identify intermittent fault behavior by observing fault behavior over time. In Adaptive Bit, the characteristics of the difference in faulty and non-

faulty behavior is used to predict whether a failed test is indicative of an intermittent or hard fault. In the sense that we are using a history of BIT results as well as information on the environment, Adaptive BIT is actually an outgrowth of Maintenance History BIT and Information Enhanced BIT.

In Temporal Monitoring BIT, the temporal behavior of faults is used to predict whether the pattern of faults seen indicates a hard fault or an intermittent fault. This is, in essence, an outgrowth and extension of Improved Decision BIT.

The problem we are concerned with can be stated as:

S1: "Given a failed BIT test (fault), what is the probability that it is intermittent?"

Alternatively, one could ask

S2: "Given a failed BIT test (fault), is it more likely that it indicates an intermittent, or a hard fault."

In Adaptive BIT, these questions are decided based on the state of the UUT and related subsystems at the time of the fault; in Temporal Monitoring, the time sequence of fault tests is used to make this decision, as described in the following subsections.

Here we use the term "fault" to indicate a failed BIT test. We will not distinguish between the failed bit test and the underlying problem that caused the failed BIT test, since our primary concern is not diagnosis. We also assume that this fault occurs repeatedly over time in an intermittent fashion and our job is to decide if the state of the UUT is either:

- OK. No fault condition occurs
- Intermittent. The fault is occuring intermittently. In section "Temporal Monitoring" this state will be split into two additional states
- Hard. A hard, permanent fault has occurred.

A schematic of how Smart BIT might be used to enhance an existing UUT is shown in Fig. E-1. The Smart BIT unit is allowed to inspect the inputs and outputs of the UUT, its BIT status, and additional sensor information that might be recorded by other sensors or the Time Stress Measurement Device (TSMD).

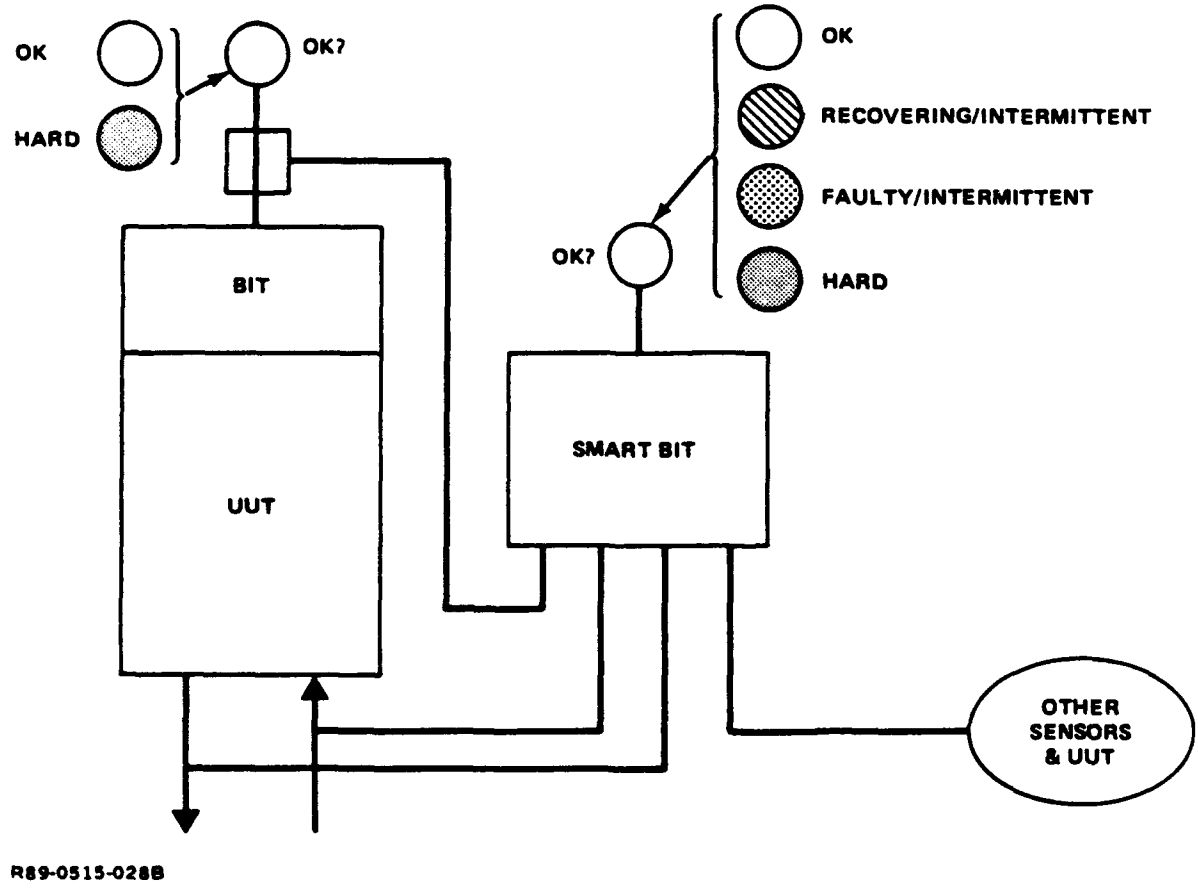



**Figure E-1. Demonstration conceptual block diagram.**

Thus for each BIT test, we can associate a vector X containing the above information. Figure E-2 depicts an example where two components of the X vector are shown, $X_o$ (temperature) and $X_1$ (vibration), for example. In that space, there is a history of previous results represented by squares for normal behavior (N) and circles for intermittent behavior (I). A new unknown behavior appears in the space as a black dot and must be properly classified as I or N.

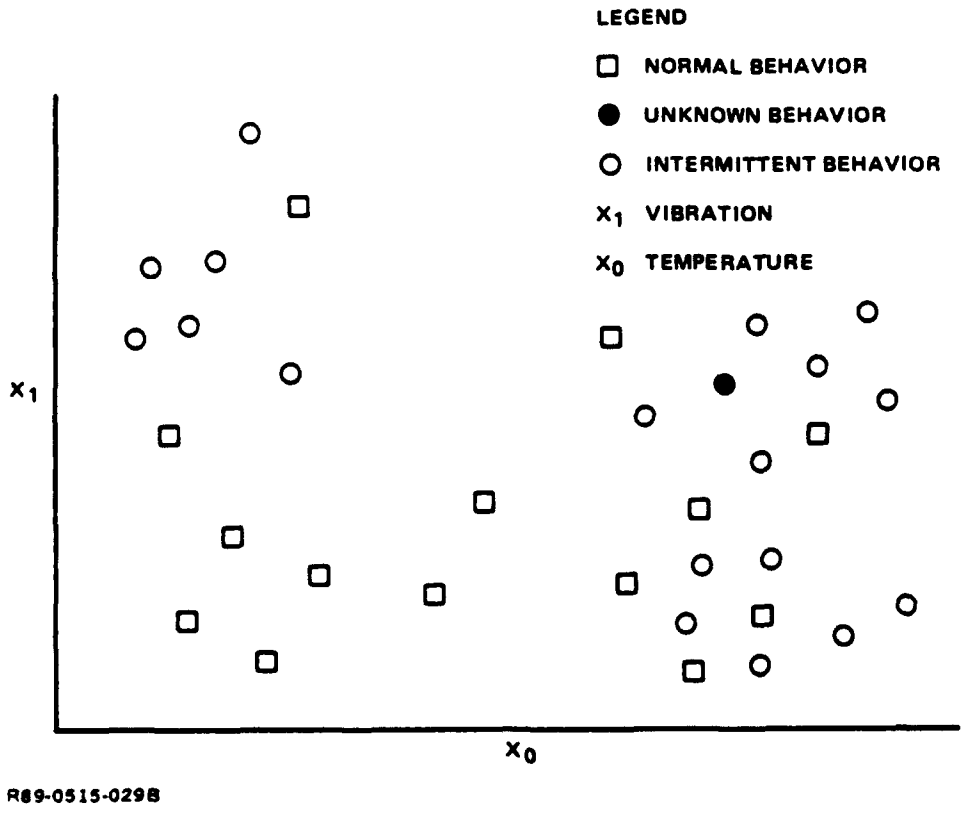



**Figure E-2. Adaptive BIT problem space.**

Given a vector X describing a new fault, we would like to determine the likelihood that

$$P(I|X) > P(H|X)$$

where $P(I|X)$ is the probability that a fault with description X is intermittent, and $P(H|X)$ is the corresponding probability that the fault is a hard fault.

Rather than estimate these probabilities directly, we will simply decide which situation, I or H, is more likelv. Thus we have a classification problem:

Given an X corresponding to a fault, classify it either as I, or H. This is a common problem in pattern recognition, and there are many ways to approach it. Two approaches have been investigated here, K-nearest Neighbors (KNN) and neural networks (NN).

Both approaches require that previously classified examples of the I and H classes are available. Examples of class I (intermittent) are those faults that have occurred previously but have been identified as intermittent because the UUT recovered. Examples of class H (Hard) are harder to come by because there is only one example, since the UUT fails to operate properly. Instead we estimate P (I|X) indirectly by assuming that

$$P(I|X) = P(OK|X)$$

where P(OK|X) is the probability that the system is operating normally. This is essentially P(X), the probability that the aircraft is in state X.

The following two subsections describe the KNN and NN approaches.

## E.1 THE K-NEAREST NEIGHBORS APPROACH TO ADAPTIVE BIT

The KNN approach is described first because it is simpler to understand. Figure E-3 shows the situation of a new fault, X, being classified. Using the K-nearest neighbor rule, X is compared to its K nearest neighbors and assigned to the class in which most of its neighbors belong. In the experiments reported here, K = 5, and 100 examples each of I and N were used.

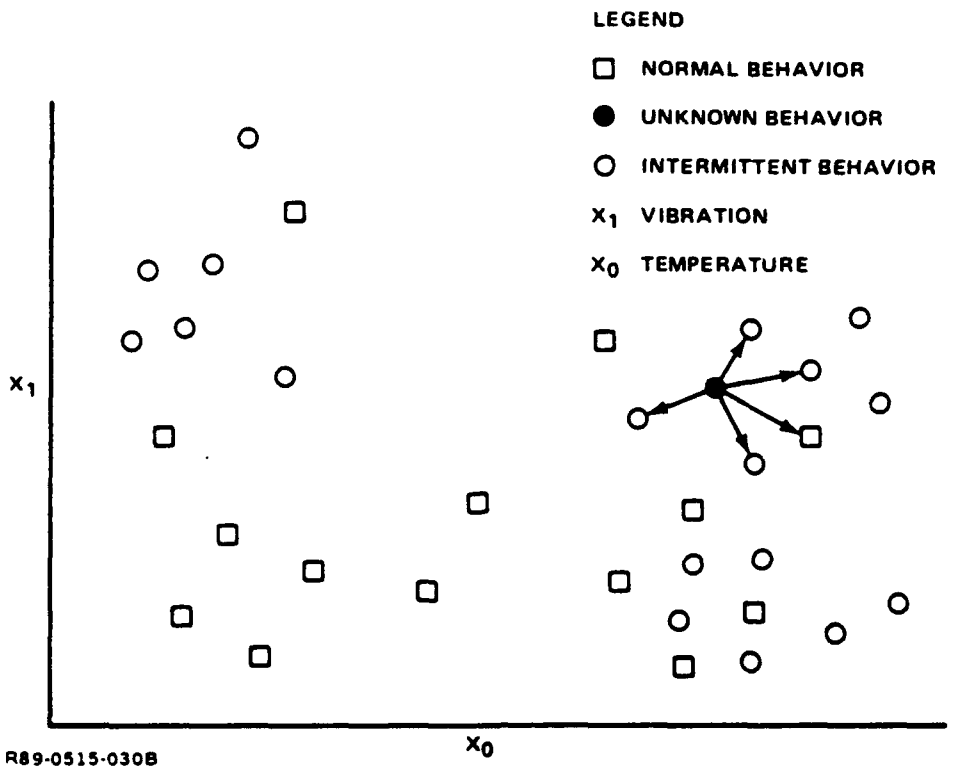



**Figure E-3. Adaptive BIT K – nearest neighbor approach.**

Basically, the K Nearest Neighbor algorithm is deciding whether P (I|X) is greater than P(H|X) or not. As K increases, the probabilities are estimated over a larger hypervolume of the X space. This increases the ability to generalize in new data, but tends to smooth the decision surface so that classification may not be perfect. Values of K near the sqrt (N) give reasonable results.

One unique aspect of the intermittent fault problem is that it has both the aspects of a classification problem and a clustering problem. It is a classification problem because one is trying to identify a point as either an "intermittent" or "normal". When these classes occur in separate regions of parameter space, we have a strict classification problem. This situation occurs, for example, when an intermittent fault always occurs when the temperature becomes high enough.

However, it may be that there is only an increased likelihood for an intermittent to occur in a particular situation. In this case, this is like a clustering or supervised learning task where one is trying to identify regions where faults tend to occur.

We approach this clustering/classification problem by first clustering intermittent and normal points and then assigning an unclassified point to the class of the cluster closest to it. The experiment discussed below was performed to evaluate the effectiveness of a KNN classifier, where K = 5, on this problem. The problem is stated in the following paragraph.

For each BIT test, we measure a parameter X with values from 0 to 1. There is a certain probability, P, of an intermittent fault occurring when X is > 0.5. The KNN classifier uses 50 examples each of the intermittent and normal classes. Its job is then to classify 100 incoming intermittent faults. Figure E-4 shows the effect of intermittent probability, P, on the misclassification error, i.e., the fraction of faults incorrectly classified as normal. The result of each trial is marked as an "x" and the solid line shows the median of 11 trials.

The experiment shows how well the system can identify a single cluster of faults as that cluster becomes more diffuse, i.e., as P decreases. When P is near 1, the misclassification error, E, is quite small, as one would expect. The error steadily increases as P decreases, but the median error remains below 0.1 until P is less then 0.4. Even with only a 0.05 probability of faulting, the median error remains below 0.15 and is seldom worse than 0.35.

This result is extremely encouraging as it suggests that such a system for recognizing intermittent faults can be effective even when tendency to fault is relatively low. In general, of course, results

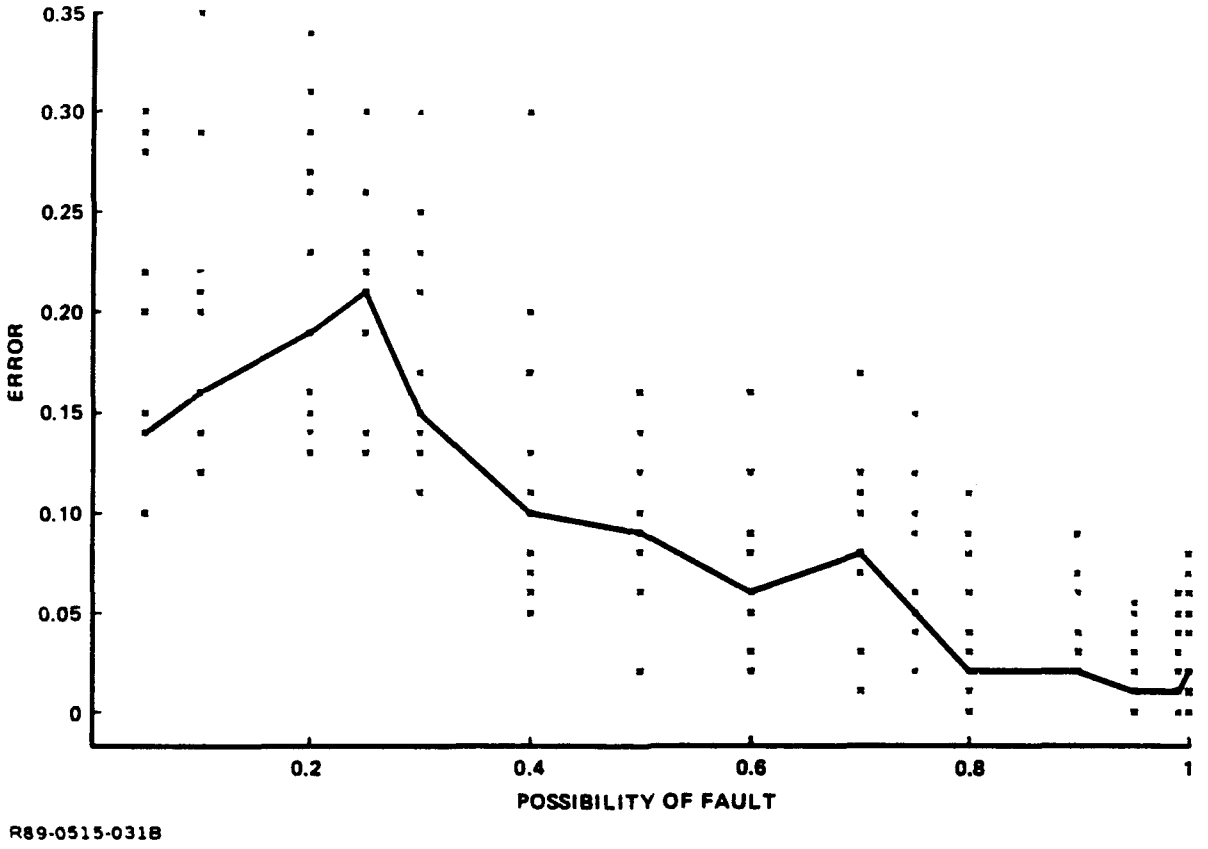



**Figure E-4. KNN classification error vs possibility of fault.**

depend on the actual number of fault clusters, C, and the number of points used in the KNN classifier, n. Also, as the fault probability is reduced, one must wait longer to acquire the necessary fault data.

## E.2 NEURAL NETWORKS

Recently, neural networks have once again become an active area of AI research. Earlier research in neural networks, during the 1950s and 1960s, led to learning algorithms for single-layer networks, called "perceptrons". Research interest in neural networks declined after Minsky and Papert showed that such networks could not compute many simple but useful functions, such as XOR. Recently, interest in neural

networks has been rekindled by the development of learning algorithms for multilayer networks with feedback.

Neural networks are being applied in two very broad classes of problems, pattern recognition and optimization. Pattern recognition tasks, as in image or speech processing, rely on the ability to either learn associations between given input and output patterns, or identify correlated patterns between input variables. Optimization problems, such as linear programming, or the traveling salesman problem, rely on the ability to rapidly reach near-optimum solutions to a problem. Neural networks can not only solve such important problems, but they can, potentially, do so blindingly fast because of their inherently parallel nature.

Neural networks are attractive in BIT applications because they can learn, and they can be implemented compactly in hardware.

### E.2.1 Neural Network Concepts

Typically, neural networks are arranged in layers, such as the three-layer network shown in Fig. E-5. The bottom layer is called the "input" layer, and it is where inputs from the outside world come in. The top layer is called the "output" layer and represents the output or answer produced by the neural network. Layers in between are called "internal" or "hidden" layers.

In most neural network research in AI, a simplified model of a biological neuron is used, such as the one shown in Fig. E-6, which shows the Jth neuron in some layer. It is connected to a set of input neurons with outputs, $X_1$ to $X_4$. Each connection between neuron i and neuron j has a corresponding weight, $w_{ij}$. The output of the jth neuron, $V_j$, is determined by the equations shown in the figure. The value $U_j$ is the weighted sum of the inputs, and the function S is a nonlinear amplifica-

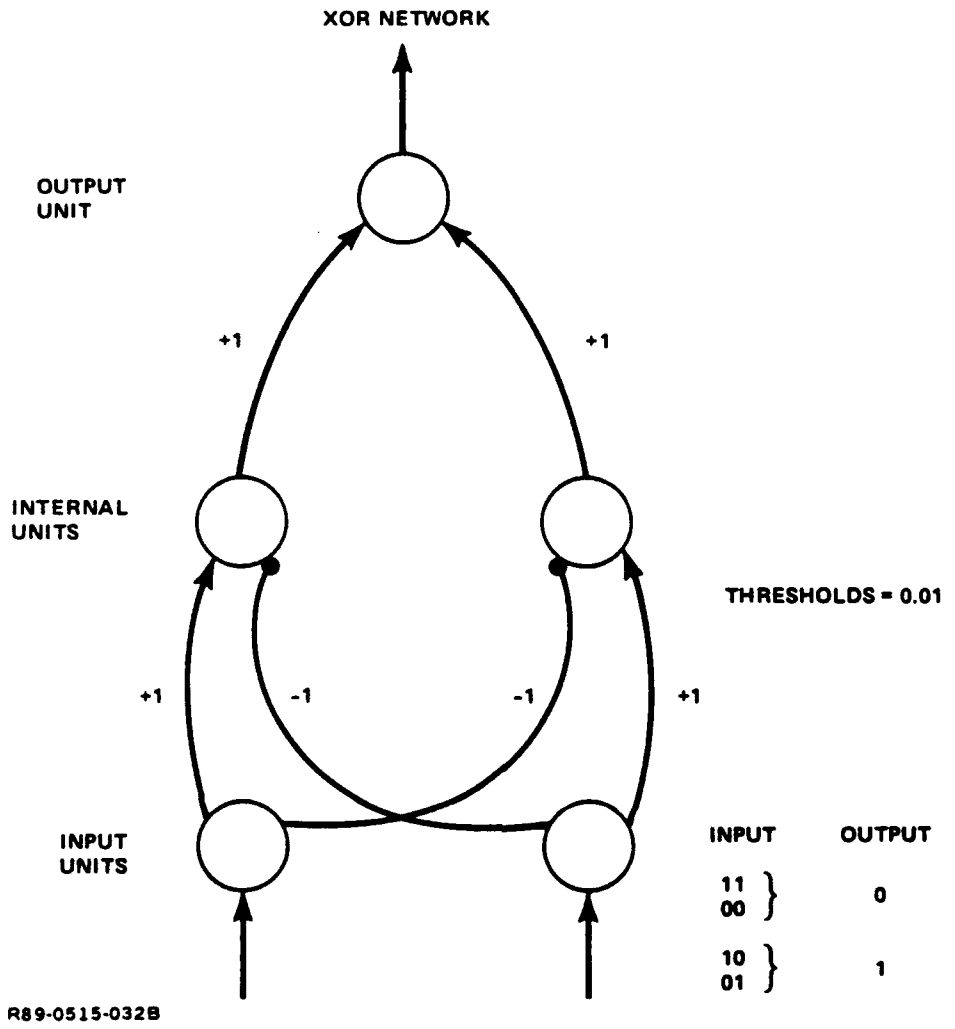
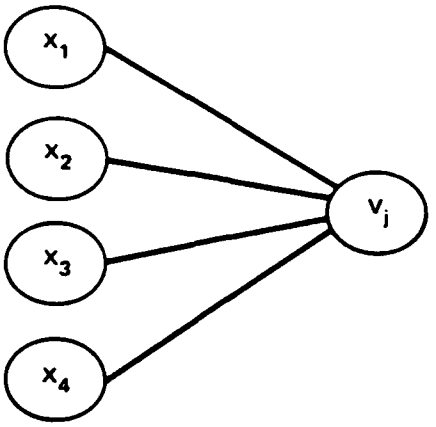



**Figure E-5. A network of linear threshold units capable of responding as an XOR.**

tion function which must be step or "S" shaped. Basically, a neuron's output is near 1 when the weighted sum of its inputs is large and near 0 when the sum is small.

The logistic function, a commonly used S function, is shown in Fig. E-7. The parameter T is used to sharpen the "S" shape. When T = 1, the curve is relatively smooth; when T = 0 the curve is a step function. During learning, T can be adjusted from a large value to progressively smaller values. Once learning is complete, T = 0 can be used.



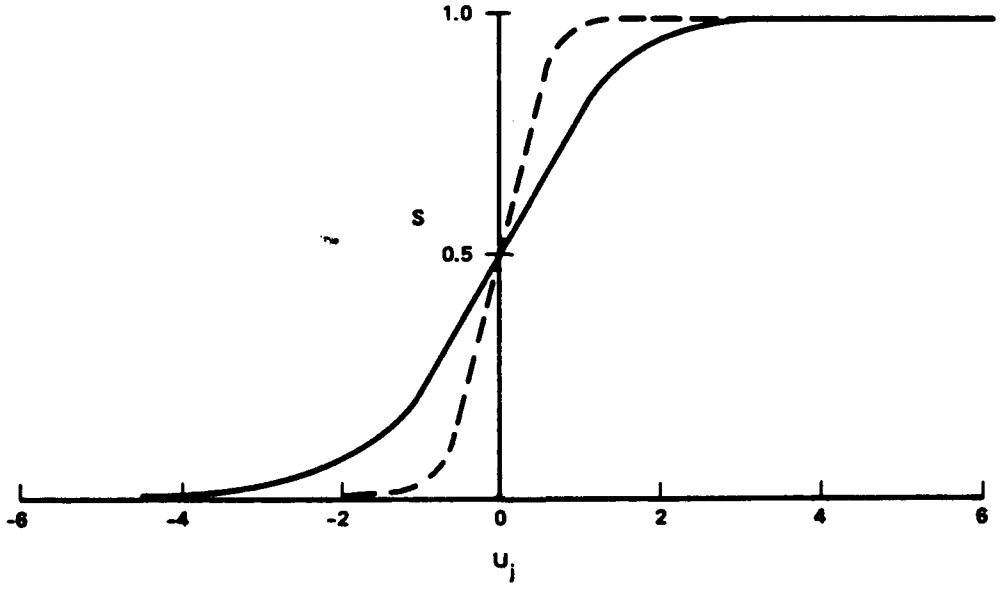

$$V_j = S(U_j)$$

$$U_j = \sum_i W_{ij} X_i$$

$$S(u) = \frac{1}{1+e^{-u/T}}, \quad 0<T<1$$

R89-0515-033B

**Figure E-6. A simplified model of a biological neuron.**




R89-0515-034B

**Figure E-7. A commonly used "S" function.**

Mathematically, for a given set of weights, the output of a single neuron defines a hyperplane in the space of its inputs. Thus, a single neuron can be used to define a linear separation between two classes.

A single layer of neurons can separate the space of inputs into two convex regions. To define more complex regions, multiple hidden layers must be used. Figure 28 shows a three-layer network that computes the XOR function of two binary inputs.

Layered networks of this kind can be implemented with extremely simple, highly parallel hardware once a set of weights has been determined. All that is required is a series of matrix vector products followed by a thresholding operation. For example, given an input vector, x, the output of a three-layer network with weight matrices $W_o$, $W_1$, and $W_2$ is:

$$S(W_2\ S\ (W_1\ S\ (W_o\ X)))$$

E.2.2 The Back-Propagation Learning Algorithm

Unfortunately, while computation with neural networks can be extremely fast, learning algorithms for neural networks can be slow. A neural network learning algorithm is an algorithm for deriving a set of connection weights for each neuron so that the output of the network solves the particular problem at hand. A commonly used learning algorithm, known as Back-Propagation, was developed by Rumelhard, Hinton, and Williams. It generalizes the perceptron learning algorithm, that only worked with one layer networks, to work with multilayer networks and is a variation of the Steepest Descent algorithm used in function optimization.

Briefly, the Back-Propagation algorithm is a form of tutored or supervised learning. Starting with a set of small random weights, the network is repeatedly shown a set of input/output pairs that it is

expected to learn. When the network is given an input, it propagates it through the network to produce an output. This output is subtracted from the desired output to give an error signal. This error signal is propagated backward through the network and the weights are adjusted slightly in the direction that will improve performance. We will now describe how back-propagation works for the case of the intermittent fault classification problem described above.

Given a vector x(t) which describes a fault, assign the fault to the class c(t), from the set (I,H). We will train a neural network to solve this classification task given a set of example input/output pairs:

$$(x(t),\ d(t)),\ 0 \leq t < T.$$

The neural network has connection weights between the output of the ith neuron v(i) to the input of the jth neuron given by

$$\left.\begin{array}{l} W_{ji} = w(k) \\ = 0 \end{array}\right\} \begin{array}{l} \text{if neuron i is connected to neuron j through weight w(i)} \\ \text{otherwise} \end{array}$$

(While it is convenient to use the matrix notation W when computing with neural networks, a corresponding vector of weights, w, is used to describe the learning algorithm.)

The output of each neuron $v_j$, is computed recursively by

$$v_j = S\ (\Sigma_i\ w_{ji}\ v_i)$$

given an initial vector of input neuron values, x(t). S is the neuron input/output function, which often has a sigmoidal shape. c(t) is the computed output vector of the neural network given the Tth training example.

Training the weights, w, of a neural network involves adjusting the weights to minimize an energy function, E:

$$E = \sum_t f\,[e(t)]$$

where e(t) is the error vector for the tth training example,

$$e(t) = d(t) - c(t)$$

and f(e) is an error metric. If the least squares error metric is used,

$$f(e) = 1/2\ e'e;\ \frac{df(e)}{de} = e$$

where "'" indicates transpose. The gradient of the energy surface is:

$$\frac{dE}{dw} = \sum_t \frac{de'(t)e(t)}{dw}$$

$$= -\sum_t \frac{dc'(t)e(t)}{dw}$$

$$= -\sum_t \frac{dc'(t)}{dw}\ d(t) - c(t)$$

$$= -\sum_t \delta(t)$$

When minimizing E using the steepest descent technique, one adjusts the weight by taking a small step in the direction of the negative gradient:

$$dw = -L\ dE/dw = +\ L\ \Sigma\delta(t)$$

where L is the step size. L can be decreased over each iteration to allow a global minimum to be attained.

The back-propagation algorithm for training neural networks is a variation on the steepest descent method described above. In back-propagation, E is minimized by adjusting the weights after each training example is run through the network by replacing the sum by a leaky integrator:

$$dw(t) = +L[delta(t) + M\ dw(t-1)\ ];\ dw(0) = 0;\ 0 < M < 1.$$

M is called "momentum" in the literature and controls the effective averaging interval. (Using M of the form $M = m^{1/T}$, we can think of m as a momentum which is independent of the number of training samples, T.)

The delta(t) is computed by first propagating the input, x(t), through the network to produce an output, c(t). Then the error e(t) = d(t) - c(t) is propagated back through the network to compute delta(t).

### E.2.3 Neural Nets Applied to Adaptive BIT

In the application to the BIT intermittent problem, the neural net approach is depicted in the problem space shown in Fig. E-8. The network

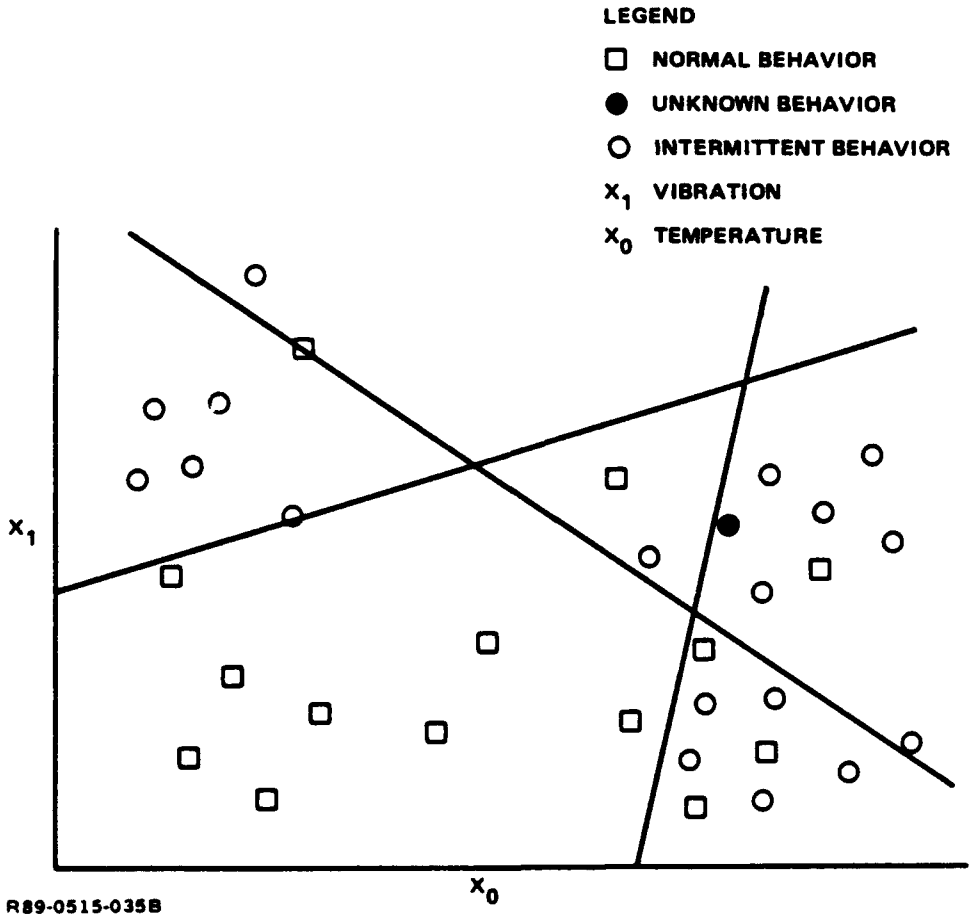



Figure E-8. Adaptive BIT neural approach.

trains on known intermittencies over a time period, adjusting its summation weights so that it properly classifies new unknown behaviors. In essence, it self-adapts to changing levels of vibration-induced faults, for example. Thus, the convex region shown breaks the space up into a normal behavior region and an intermittent behavior region, and classifies the unknown behavior (black dot) as an intermittency.

## E.3 TEMPORAL MONITORING

In Temporal Monitoring, the temporal behavior of faults is used to predict whether the observed pattern of failures indicates a hard fault, an intermittent fault, or whether an interval of intermittent behavior is likely to be over. The main advantage to temporal monitoring over the commonly used "M out of N" approach is the ability to adapt to the actual fault situation.

The "M out of N" approach can be stated as follows:

> When a BIT test fails, apply the test N more times, and if it fails M of those times, report that a fault has occurred.

The problem with this approach is that M and N are predetermined. Even if M and N are carefully chosen, they may not represent a particular fault situation. For example, suppose that M or N or both are a bit too low. Then a certain amount (possibly all) of the false alarm rate could be due to the choice of M and N alone, and not to any fault in the hardware. In an adaptive approach, M and N could be adjusted so that, after a short period of false alarms, the false alarm rate would be reduced.

Temporal monitoring uses the theory of Bernoulli random variables and Markov models to develop a temporal model of an intermittent fault. First Bernoulli random variables are reviewed, which leads to an understanding of how M and N can be chosen.

In a Markov model, at any given time, the system being modeled can be in one of a finite set of states. Transitions between states is governed by a probability on each transition. The Markov model used in this study is shown in Fig. E-9. It has four states: OK, HARD, RECOVERING, and FAULTY. The RECOVERING and FAULTY states can be thought of as two substates of an intermittent state.

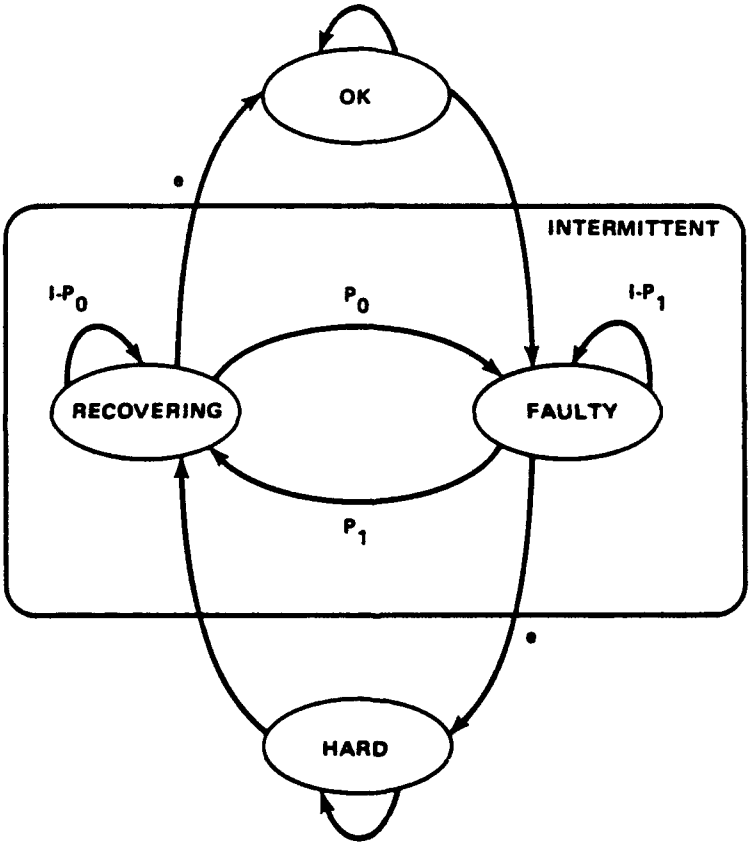



**Figure E-9. Markov model of the intermittent behavior.**

The states can be interpreted as:

- OK: The UUT is operating normally
- HARD: The UUT has a hard fault

The theory of Bernoulli random variables involves a sequence of "trials". In our case, a trial is a BIT test. Let P be the probability of an event of interest on the next trial. Thus the probability of the event not occurring on the next trial is 1-P. For example, the event of interest could be "the BIT test fails". Then the probability that K events occur in N trials is

$$P(K,N) = C(N,K)P^{K}(1-P)^{N-K}$$

The probability that the first event happens on the n'th trial is

$$PF(N) = P(1-P)^{N-1}$$

The probability that the first event will happen within K trials is

$$PK = 1-(1-P)^{K}$$

So, given P, the K for which PK > 0.95 is

$$K(\geq \text{ceiling } (\log (1 - PK) / (\log (1 - P)))$$

Alternatively, given K, the P is

$$P = 1-(1-PK)^{1/k}$$

This tells us how long we must wait for a fault to occur before we can be 95% confident that the actual probability of failing a BIT test is below a certain probability P, as shown in the following table:

| P | K |
|---|---|
| 0.5 | 5 |
| 0.1 | 29 |
| 0.01 | 299 |
| 0.001 | 2995 |

For example, if we perform a BIT test five times without a failure, we can presume that the probability of the next test failing is less than 0.5. However, if we want to be sure that the probability of the next test failing is less than 0.01, we must wait for 299 or more successful tests.

- FAULTY (INTERMITTENT): The UUT is experiencing BIT test failures due to an intermittent fault
- RECOVERING (INTERMITTENT): The UUT is not failing any BIT tests, but has not behaved normally long enough to be completely trusted.

The idea behind temporal monitoring is to use the sequence of success and failure of a BIT test to decide which of the four states the UUT is in and to estimate the transition probabilities. The probabilities are estimated using a moving average of the number of transitions on each arc. In our experiments, only $P_o$ and $P_1$ are estimated as it was assumed that the transitions (from RECOVERING to OK, and FAULTY to HARD) do not occur often enough to be reliably estimated.

The following summarizes the behavior of the monitor in each state:

- OK: The system starts in state OK and transitions to state INTERMITTENT when the BIT test fails
- FAULTY: If a BIT test succeeds, the monitor transitions to state RECOVERING. Otherwise, if the number of consecutive failed tests is high enough, a transition to state HARD occurs. This condition is based on the PK probability described above and the $P_1$ being estimated dynamically
- RECOVERING: Similarly, if a BIT Test fails, the monitor transitions to state FAULTY. Otherwise, if the number of consecutive successful tests is high enough, a transition to state OK occurs. This condition is also based on the PK probability and the $P_o$ being estimated dynamically

- HARD: If a successful BIT test occurs, a transition to state RECOVERING occurs.

The monitor always provides an estimate of the health of the UUT. The initial values of $P_o$ and $P_1$ are chosen to be reasonably small probabilities, such as 0.001. Then, as intermittent faults occur, these values are adapted to appropriate values.

The additional state information provided by the monitor lets BIT software make softer decisions. How this information is used depends upon the UUT. For example, if a UUT is determined to be INTERMITTENT and it is redundant, than its information can be ignored until it is considered OK again. Alternatively, if the UUT is absolutely critical, knowing that it is RECOVERING may be enough to save a mission.

# APPENDIX F

# SMART BIT/TSMD INTEGRATION SYSTEM SCENARIO SCRIPTS

## F.1 SMART BIT/TSMD SYSTEM SETUP

The instruction steps to follow (subsections and beyond) are set up to provide an explanation of the step first and then the actual instruction(s) (simple steps combine both explanation and instruction). Steps that require multiple instructions have each instruction numbered. Perform the instructions for each step in the order shown. A note about the instruction may also follow directly below it. An options note may follow the instruction to indicate different uses for it.

The following explains the entry of certain key sequences showing the scenario scripts:

| Key Sequence: | How to Enter at Keyboard: |
|---|---|
| **[Enter]** | Press the Enter key only (also referred to as carriage return). |
| **[Esc]** | Press the Escape key only. |
| **[Shift]-[>]** | Press and hold down the Shift key, and then press the > key. |
| **[Ctrl]-[s]** | Press and hold down the Ctrl key (Control key), and then press the s key. |
| **[Pg Up]** | Press the Pg Up key only (located in the number keyset area on the right side of the keyboard). |
| **[Alt]-[F4]** | Press and hold down the Alt key (Alternate key), and then press the F4 key. |
| **[Alt]-[n]** | Press and hold down the Alt key (Alternate key), and then press the n key. |
| **[Ctrl]-[g]** | Press and hold down the Ctrl key (Control key) and then press the g key. |
| **[Ctrl]-[c]** | Press and hold down the Ctrl key (Control key) and then press the c key. |

The **[Ctrl]-[g]** sequence can be used on the Smart BIT computer to abort LISP "do-scenario" function (and most other LISP functions). The **[Esc]** key may be used to abort certain Microsoft Windows menu operations on either the UUT or TSMD computers.

When an instruction refers to clicking the mouse button, please note the number of times this is to be done (ONCE or TWICE). Clicking ONCE means depressing the selected mouse button and letting go only once. This is usually done to acknowledge a request to perform some task (or to abort or stop a task). Clicking TWICE means to depress the selected mouse button and let go two times in moderately rapid succession. This operation usually is used to open file or start up a process. Be sure to maintain the mouse cursor over the place indicated while performing a click operation.

It should be noted that there are at least two types of cursors. The arrow cursor is moved by the mouse and is used to select something in a dialog box, menu window, or graphic tool. The text cursor is a rectangle (or a flashing vertical line in the case of the UUT Scenario Generation Tool Editor) and indicates the current position of the next character to be typed in.

Since the Test Vehicle software's simulated (the application program running on the LRU BIT computer) LRU behavior is set up and controlled by the UUT computer, and the Test Vehicle software automatically starts up when the LRU BIT computer is turned on. Therefore, no LRU BIT computer or Test Vehicle software operations need to be performed by the user (except powering the LRU BIT computer on or off ) to initiate any scenario script sequences.

**WARNING: THE SMART BIT COMPUTER RUNS THE UNIX OPERATING SYSTEM. ANY COMPUTER RUNNING THE UNIX OPERATING SYSTEM MUST NOT BE POWERED DOWN WHILE STILL IN AN APPLICATION PROGRAM. FOLLOW THE INSTRUCTIONS IN SECTION F.3 FOR THE PROPER POWER SHUTDOWN SEQUENCE (THE UNIX OPERATING SYSTEM USUALLY WILL "REPAIR" ANY DISK DATA DAMAGE AUTOMATICALLY, BUT POSSIBLE DISK DATA LOSS CAN OCCUR IF THE COMPUTER IS NOT POWERED DOWN PROPERLY AND LARGE AMOUNTS OF DATA WAS LOST).**

### F.1.1 Standard Conditions

All operations will be performed under the following standard conditions:

- Temperature.....20 ± 10°C
- Altitude.....Normal ground
- Vibration.....None
- Humidity.....Room ambient
- Power Requirements.....Normal facility 60 Hz power, 115 + 10%, 60 Hz + 3 Hz, single-phase.

NOTE: The following are the reference names used herein with their identifications:

- UUT computer — Z-248 with monochrome monitor
- LRU BIT computer — CS/5 chassis, no monitor
- TSMD computer — Z-248 with color monitor
- Smart BIT computer — Compaq 386 with high-resolution color monitor.

### F.1.2 Power Up The Smart BIT/TSMD Integration Test Bed Computers

The Smart BIT/TSMD Integration System must be interconnected according to the interconnect diagram, Figure F-1, before switching on any computers.

On all computers and monitors, depress power ON/OFF switches to the ON position in following order: LRU BIT computer, TSMD computer, UUT computer, Smart BIT computer then perform the following steps.

1. Observe that the Power-On green or red lamp on the front of the each computer chassis and/or monitor is illuminated.
2. Adjust the brightness control (if necessary) to make each computer display visible.
3. At the Smart BIT Computer, Log in and enter the password as shown below (this starts up the Smart BIT software):

   **root[Enter]**

   **daler1[Enter]**

NOTE 1: If a CRON message appears on the Smart BIT computer screen login line, ignore it and type (wherever the cursor appears) the above information exactly as shown.

NOTE 2: There is a time-out on the Smart BIT computer display which blanks out the screen if there is no key or mouse activity for about 30 minutes. If this occurs during normal operation, move the Smart BIT computer mouse a small distance to restore (unblank) the Smart

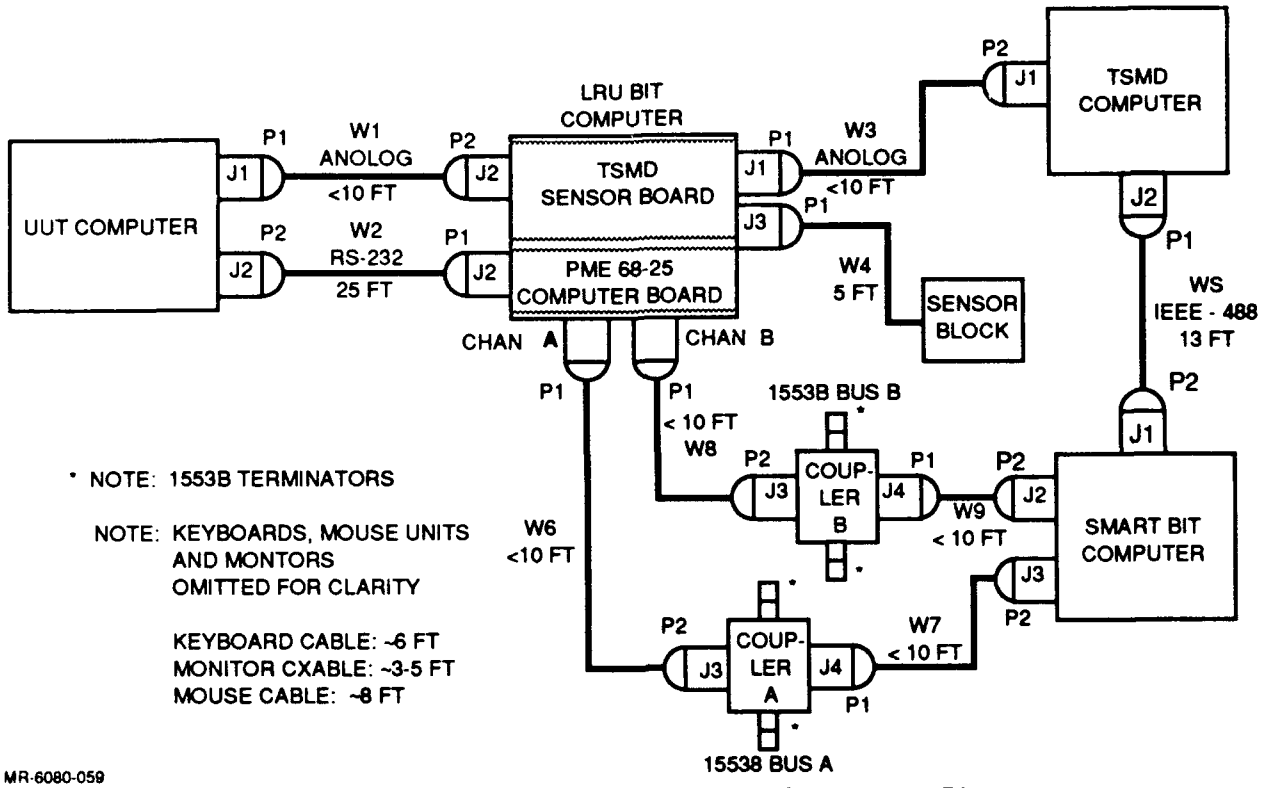



**Fig. F-1 Smart Bit/TSMD Integration System Interconnect Diagram.**

BIT computer display. Be sure to maintain the mouse cursor somewhere over the Smart BIT LISP LISTENER window.

4. While the Smart BIT programs are loading, ignore all warning messages. Allow several minutes for the Smart BIT programs to load and then continue with the next step.
5. Acknowledge the TSMD computer software restricted rights window:
   A. Move the TSMD computer mouse onto the TSMD computer restricted rights window over the box surrounding the word:
      OK
   B. Click the left button on the TSMD computer mouse ONCE.
6. Acknowledge the TSMD Self-Test passed window:
   A. Move the TSMD computer mouse onto the TSMD computer Self-Test window, over the box surrounding the word:
      OK
   B. Click the left button on the TSMD computer mouse ONCE.
7. Select the option to load a configuration file that contains an x,y positioning of TSMD data windows:

A. Move the TSMD computer mouse onto the Configuration window, over the box surrounding the word:

YES

B. Click the left button on the TSMD computer mouse ONCE.

8. Select the actual configuration file that contains an arrangement of TSMD windows:

A. Move the TSMD computer mouse onto the Configuration file list window, over the word:

TEST.CFG

B. Click the left button on the TSMD computer mouse TWICE.

NOTE: the TSMD windows saved under TEST.CFG recreates the windows as they were when the program was last exited. As such, the present window should agree with Figure F-3 in format but not in exact content.

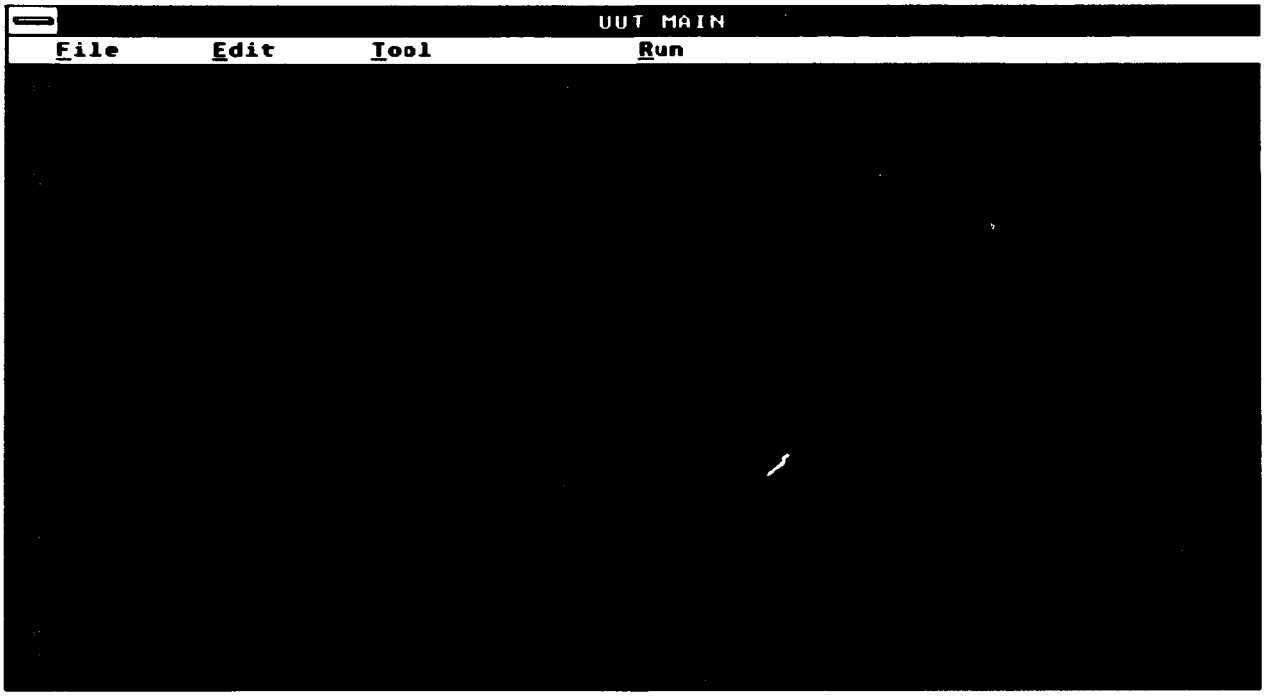
9. Acknowledge the UUT self-test passed:

A. Move the UUT computer mouse onto the UUT computer self-test window, over the box surrounding the word:

OK

B. Click the left button on the UUT computer mouse ONCE.

10. Observe that the UUT, TSMD and Smart BIT computer displays are as shown in Figures F-2, F-3, and F-4.




R91-6060-060

**Figure F-2. UUT interface window.**

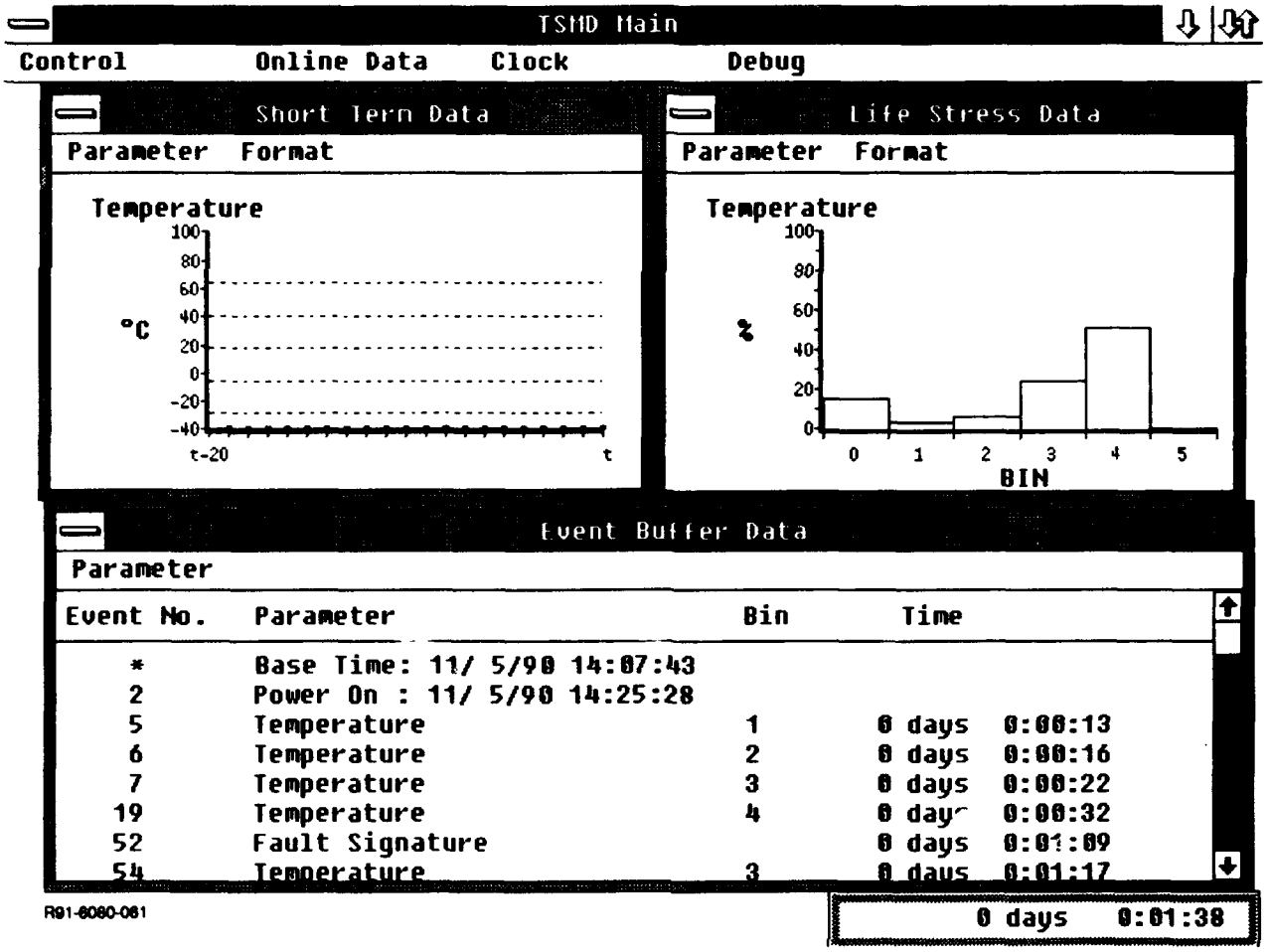



**Figure F-3. Initial TSMD window.**

### F.1.3 Smart BIT/TSMD Integration System Test

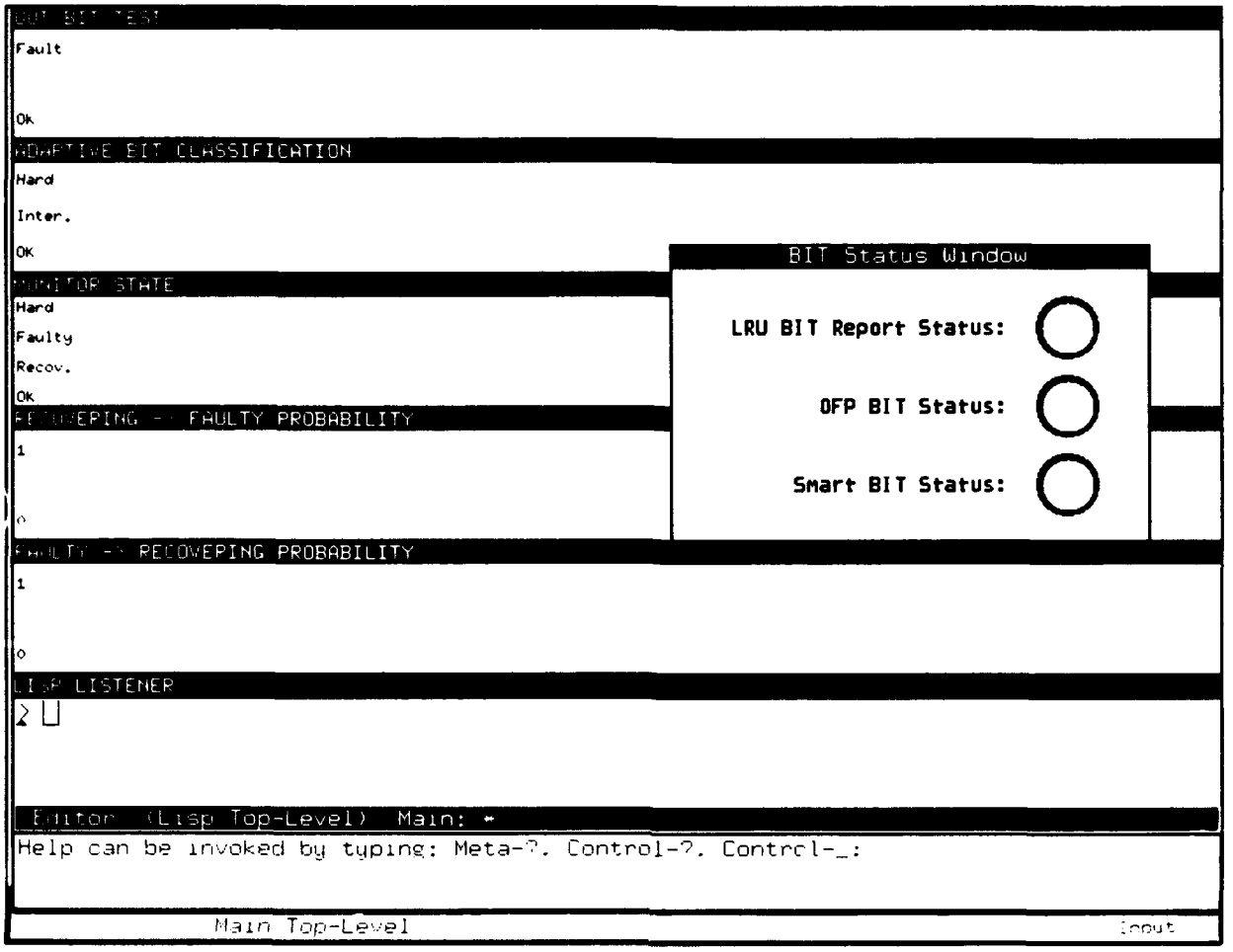
1. Type the following into the LISP LISTENER window of the Smart BIT Computer:

   **(system-test)**

2. At the completion of test execution, observe that the system test message on the Smart BIT computer LISP LISTENER window reads as follows:

   Smart BIT - OK TSMD - OK LRU BIT - OK UUT - OK

NOTE: The above message does not appear as shown above if one or more computers indicates a FAULT. The next text line should identify the specific problem of one of the faulted computers. Usually it is a NO RESPONSE type fault. This means that either the computer is not turned on, or it has a bad cable connection or is disconnected, and therefore is not responding, or that the applications program is not running and therefore cannot respond to the system test message.




R91-6080-062

**Figure F-4. Initial Smart BIT window.**

## F.2 INSTRUCTIONS FOR RUNNING SCENARIOS

### F.2-1 Initializing the Smart BIT and TSMD Computers

1. Reset the Smart BIT text cursor position to be visible on the Smart BIT LISP LISTENER window:

At the Smart BIT computer, type the following:

**[Esc]**

**[Shift]-[>]**

**(bit-set)[Enter]**

2. Observe that the selections in the Smart BIT Parameters menu window are as follows: (wait for the Smart BIT Parameter menu window to appear):

Display BIT Status Window: YES

Display Average Error Window : NO

3. Save the selections for the Smart BIT Parameters menu window:
   A. Move the Smart BIT mouse cursor (arrow) onto the the bottom left Smart BIT Parameter menu window over the word:
      ENTER
   B. Click the Smart BIT computer left mouse button ONCE.
4. Start the Smart BIT software to run a scenario. Type the following:
   **(do-scenario "" :control :bus :learn-by :neural-net)**

NOTE: The do-scenario function takes some other options for the control keyword. The :bus keyword uses the MIL-STD-1553B and IEEE-488 buses for the source of data. The two other control keyword options that may be used instead of the :bus option are the :write-file option (same as :bus option but saves data to a file specified in the double quotes) and the :read-file option (reads data from the file specified in the double quotes instead of the MIL-STD-1553B and IEEE-488 buses). The :learn-by keyword option also has a :knn (for K-nearest neighbor Adaptive BIT) option in addition to the :neural-net option shown.

Example 1: (do-scenario "/lisp/test" :control :write-file :learn-by :knn)

This command line reads in the MIL-STD-1553B and IEEE-488 bus data, writes the file called "test" (in the /lisp directory) with the MIL-STD-1553B and IEEE-488 bus input data, and then analyzes the bus data using the K-nearest neighbor Adaptive BIT technique.

Example 2: (do-scenario "/lisp/test" :control :read-file :learn-by :knn)

This command line reads in the file called "test" (in the /lisp directory) as the input data and then analyzes this file data using the K-nearest neighbor Adaptive BIT technique. The :read-file keyword option does not use any external buses to obtain the input data since this information is supplied by the file and therefore does not require any of the other Smart BIT / TSMD Integration computers to run a scenario. The :write-file command line option must first be run in order to take advantage of the :read-file scenario replay option.

5. At the TSMD Computer, display the TSMD System Control Dialog box by typing the following:
   **[Ctrl]-[s]**

6. Select to reset all previous TSMD data in memory:
   A. Move the TSMD computer mouse cursor (arrow) onto the TSMD computer system Control Dialog box Memory Resets group area and then over the box surrounding the word:
      ALL
   B. Click the TSMD computer left mouse button ONCE.
7. Extinguish the System Control Dialog box:
   A. Move the TSMD computer mouse cursor (arrow) onto the the bottom left of the System Control Dialog box over the box surrounding the word:
      OK
   B. Click the TSMD computer left mouse button ONCE.

NOTE: Observe that the TSMD display appears similar to Figure F-5.

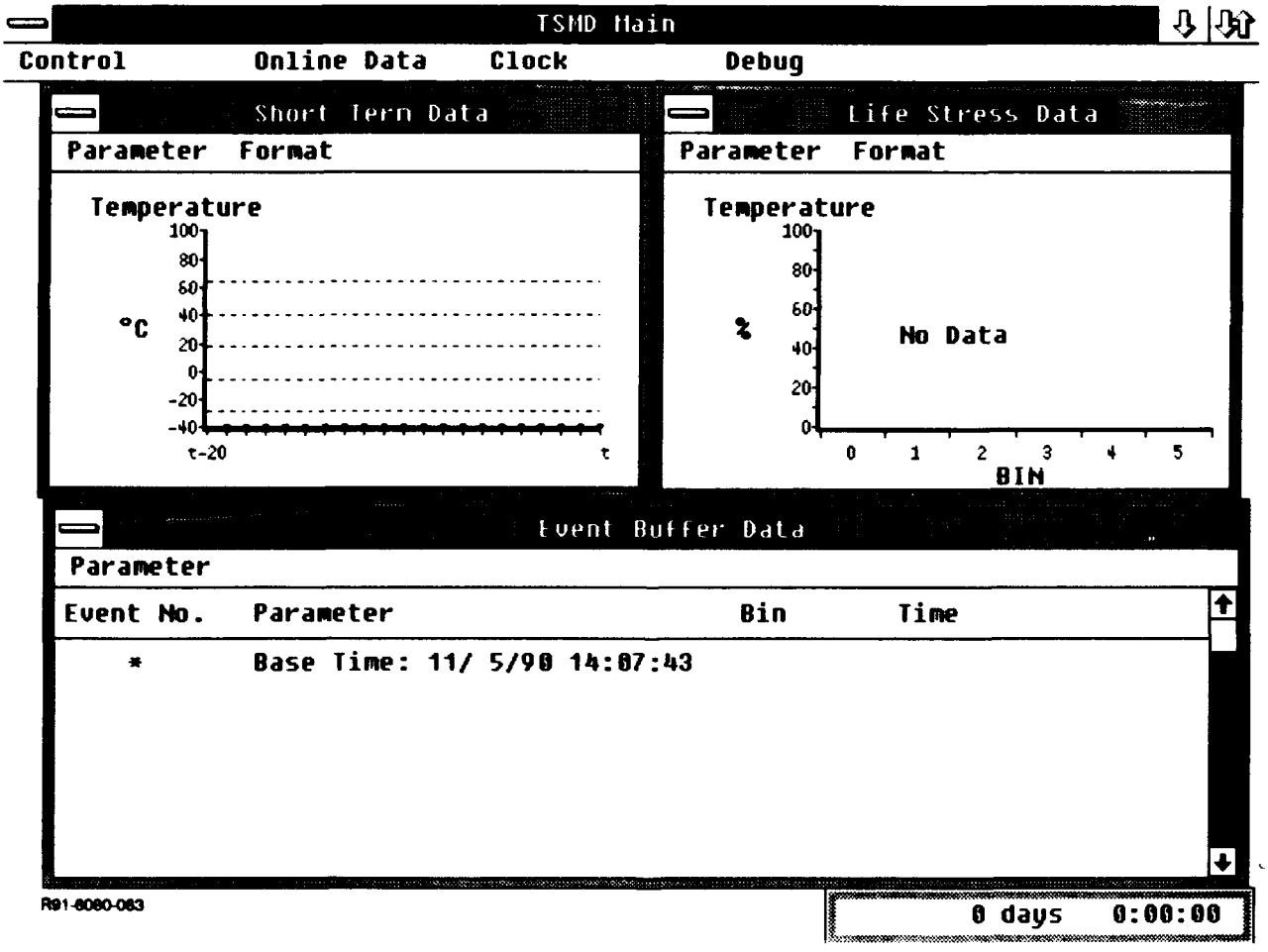



Figure F-5. TSMD window reset

### F.2.2 Execute Scenario No. 1

1. Setup the UUT computer to run a scenario:

A. Move the UUT computer mouse cursor (arrow) to the menu bar (near the top of the UUT Main window) over the word:

Run

B. Click the UUT computer left mouse button ONCE.

2. Wait for the Run Dialog box to appear. Now select scenario no. 1 and run it:

A. Move the UUT computer mouse cursor (in the file directory list of the Run Dialog box) over the scenario no.1 filename:

SCEN1.RUN

B. Then click the UUT computer left mouse button TWICE.

3. Wait for small UUT window to appear at the center of the UUT computer screen (it will say: Running Scenario SCEN1.RUN). Scenario no. 1 is now running.

OPTION NOTE: Once running, if it is desired to stop the scenario before its normal end (abort), move the UUT computer mouse onto the small UUT window over the box surrounding the word: Cancel. Then click the UUT computer left mouse button ONCE.

4. Observe the following results at the TSMD computer:

A. Observe that when the scenario begins, the Life Stress Data window display begins changing. The elapsed time displayed in the lower right corner of the screen will begin counting up, and a power-up event will be logged in the Event Buffer Data window.

B. Observe that when the temperature trace displayed in the Short Term Data window crosses one of the thresholds drawn in red, an event is logged in the Event Buffer Data window.

C. Observe that the bar graphs displayed in the Life Stress Data window change in accordance with the data displayed in the Short Term Data window. For example, when the temperature trace is in range 3, the bar for that range grows while the others shrink.

D. When the scenario ends, note that activity in the Life Stress Data and Event Buffer Data windows stops. The display should resemble that shown in Figure F-6.

5. Observe the following results at the Smart BIT computer:

A. At the end of the data analysis, observe that the window displayed agrees with the typical window shown in Figure F-7.

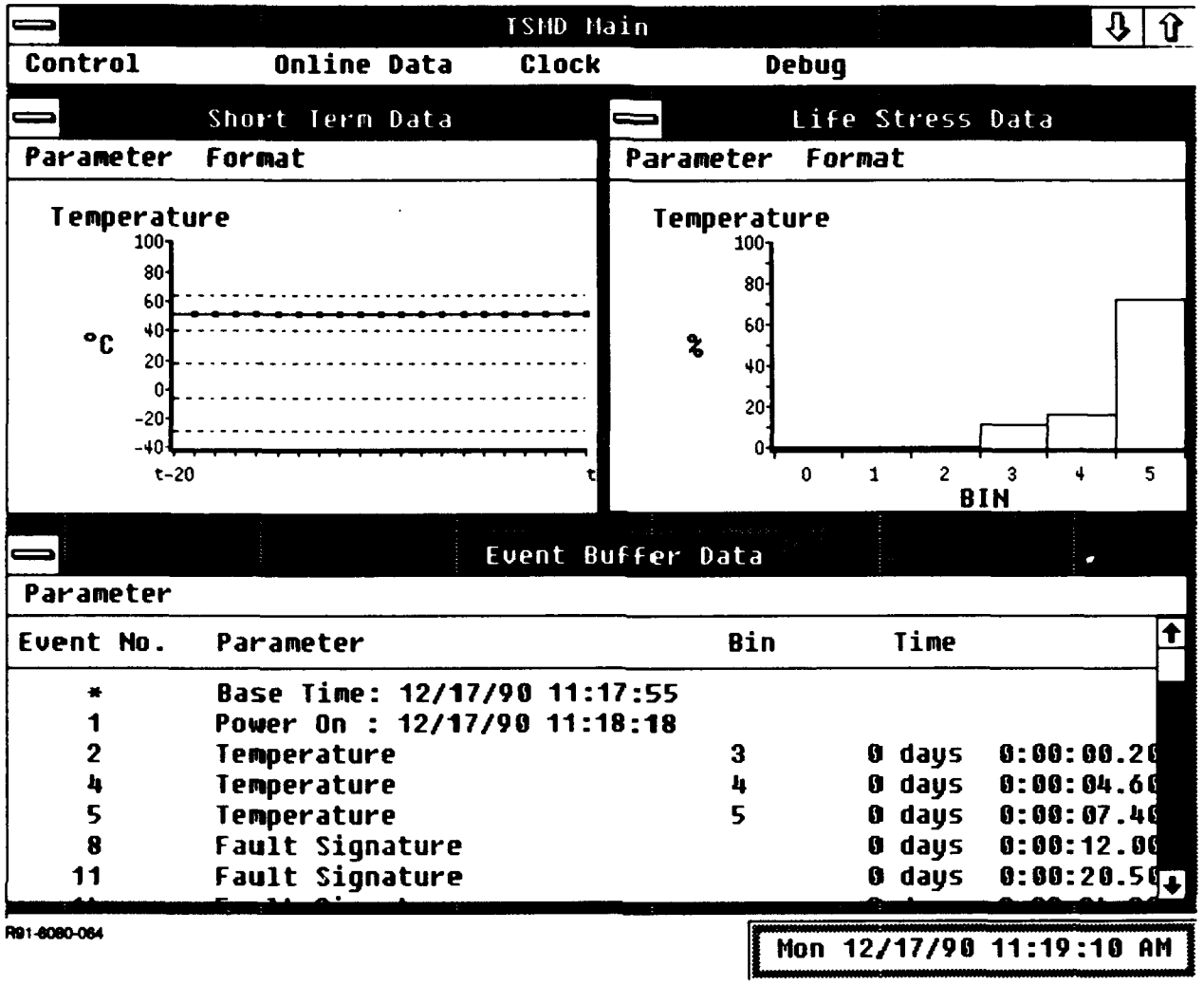



**Figure F-6. Scenario 1 TSMD window.**

NOTE: Because of the asynchronous nature of the Smart BIT / TSMD Integration System test bed, there may be differences between Figure G-7 and the actual window results.

B. Using the **[Pg Up]** key, observe that the LISP LISTENER window contains the following:

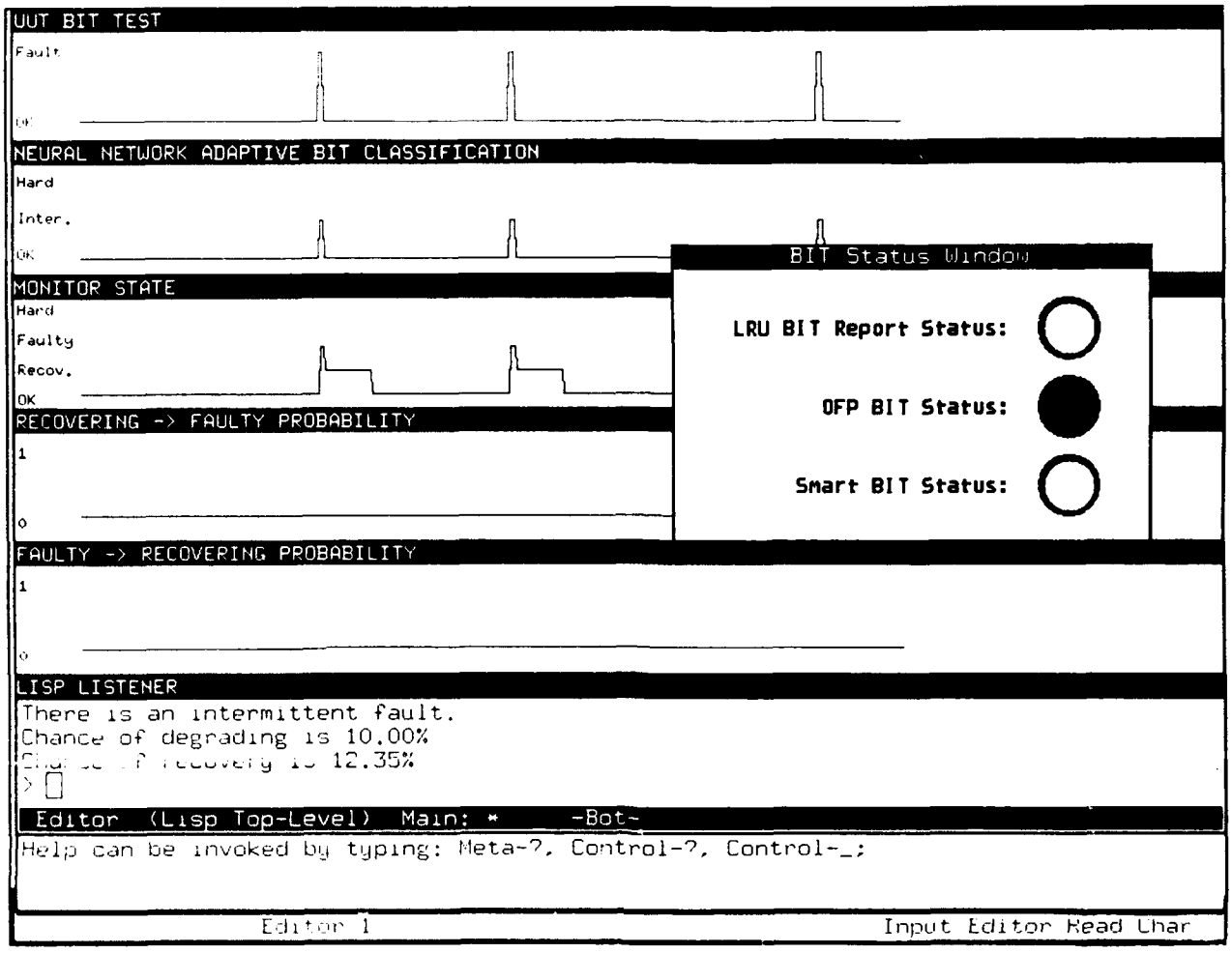
0 Faulty Samples Misclassified.

There is an intermittent fault.

Chance of degrading is 10.00 %.

Chance of recovery is 12.20%.

NOTE: Because of the asynchronous nature of the Smart BIT / TSMD Integration System test bed, there may be differences in percentages displayed versus those shown above. There may also be some differences in the display data.




R91-6080-065

**Figure F-7. Scenario 1 Smart BIT window.**

## F.2.3 Execute Scenario No. 2

1. Reposition the Smart BIT LISP LISTENER text cursor back to its proper place. At the Smart BIT Computer keyboard, type the following:

   **[Esc]**

   **[Shift]-[>]**

2. Re-display the last command on the Smart BIT LISP LISTENER window. At the Smart BIT computer keyboard, type the following:

   **[Esc]**

   **[p] [ENTER]**

3. Remove the temperature data from the TSMD computer Short-Term window:

   A. At the TSMD computer, move the TSMD computer mouse cursor onto the TSMD

Computer Short Term Data window over the word:

Parameter

B. Click the TSMD computer left mouse button ONCE.

C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:

Temperature

D. Click the TSMD computer left mouse button ONCE.

4. Place the Vibration Peak data onto the TSMD computer Short-Term window:

A. Move the TSMD computer mouse cursor onto the TSMD Short-Term window over the word:

Parameter

B. Click the TSMD computer left mouse button ONCE.

C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:

Vibration Peak

D. Click the TSMD computer left mouse button ONCE.

5. Remove the temperature data from the TSMD computer Life Stress Data window:

A. At the TSMD computer, move the TSMD computer mouse cursor onto the TSMD computer Life Stress Data window over the word:

Parameter

B. Click the TSMD computer left mouse button ONCE.

C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:

Temperature

D. Click the TSMD computer left mouse button ONCE.

6. Place the Vibration Peak data onto the TSMD Life Stress window:

A. Move the TSMD computer mouse cursor onto the TSMD Life Stress window over the word:

Parameter

B. Click the TSMD computer left mouse button ONCE.

C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:

Vibration Peak

D. Click the TSMD computer left mouse button ONCE.

7. Remove the temperature data from the TSMD computer Event Buffer Data window:
   A. At the TSMD computer, move the TSMD computer mouse cursor onto the TSMD Computer Event Buffer Data window over the word:
      Parameter
   B. Click the TSMD computer left mouse button ONCE.
   C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:
      Temperature
   D. Click the TSMD computer left mouse button ONCE.
8. Place the Vibration Peak data onto the TSMD Event Buffer window:
   A. Move the TSMD computer mouse cursor onto the TSMD Event Buffer window over the word:
      Parameter
   B. Click the TSMD computer left mouse button ONCE.
   C. Move the TSMD mouse cursor onto the Parameter menu and place mouse cursor over the menu option word:
      Vibration Peak
   D. Click the TSMD computer left mouse button ONCE.
9. At the TSMD computer, display the TSMD System Control Dialog body by typing the following:
   **[Ctrl]-[s]**
10. Select to reset all previous TSMD data in memory:
   A. Move the TSMD computer mouse cursor (arrow) onto the TSMD computer System Control Dialog box Memory Resets group area and then over the box surrounding the word:
      ALL
   B. Click the TSMD computer left mouse button ONCE.
11. Extinguish the System Control Dialog box:
   A. Move the TSMD computer mouse cursor (arrow) onto the the bottom left of the System Control Dialog box over the box surrounding the word:
      OK
   B. Click the TSMD computer left mouse button ONCE.
12. Set Up the UUT computer to run a scenario:
   A. Move the UUT computer mouse cursor (arrow) onto the menu bar (near the top of the

UUT Main window) over the word:

Run

B. Click the UUT computer left mouse button ONCE.

13. Wait for the Run Dialog box to appear. Now select scenario no. 2 and run it:
    A. Move the UUT computer mouse cursor (in the file directory list of the Run Dialog box) over the scenario no.1 filename:

       SCEN2.RUN

    B. Click the UUT computer left mouse button TWICE.
14. Wait for small UUT Main Window to appear at the center of the UUT computer screen (it will say: Running Scenario SCEN2.RUN). Scenario no. 2 is now running.

OPTION NOTE: Once running, if it is desired to stop the scenario before its normal end (abort), move the UUT computer mouse onto the small UUT Main Window over the word: Cancel. Then click the UUT computer left mouse button ONCE.

15. Observe the following results at the TSMD computer Display:
    A. Observe that when the scenario begins, the Life Stress Data window display begins changing. The elapsed time displayed in the lower right corner of the screen will begin counting up, and a power-up event will be logged in the Event Buffer Data window.
    B. Observe that when the Vibration Peak trace displayed in the Short-Term Data window crosses one of the thresholds drawn in red, an event is logged in the Event Buffer Data window.
    C. Observe that the bar graphs displayed in the Life Stress Data window change in accordance with the data displayed in the Short-Term Data window. For example, when the Vibration Peak trace is in range 3, the bar for that range grows while the others shrink.
    D. When the scenario ends, note that activity in the Life Stress Data and Event Buffer Data windows stops. The display should resemble that shown in Figure F-8.
16. Observe the following results at the Smart BIT computer Display:
    A. At the end of the data analysis, observe that the window displayed agrees with the typical window shown in Figure F-9.

NOTE: Because of the asynchronous nature of the Smart BIT / TSMD Integration system test bed, there may be differences between Figure F-9 and the actual window results.

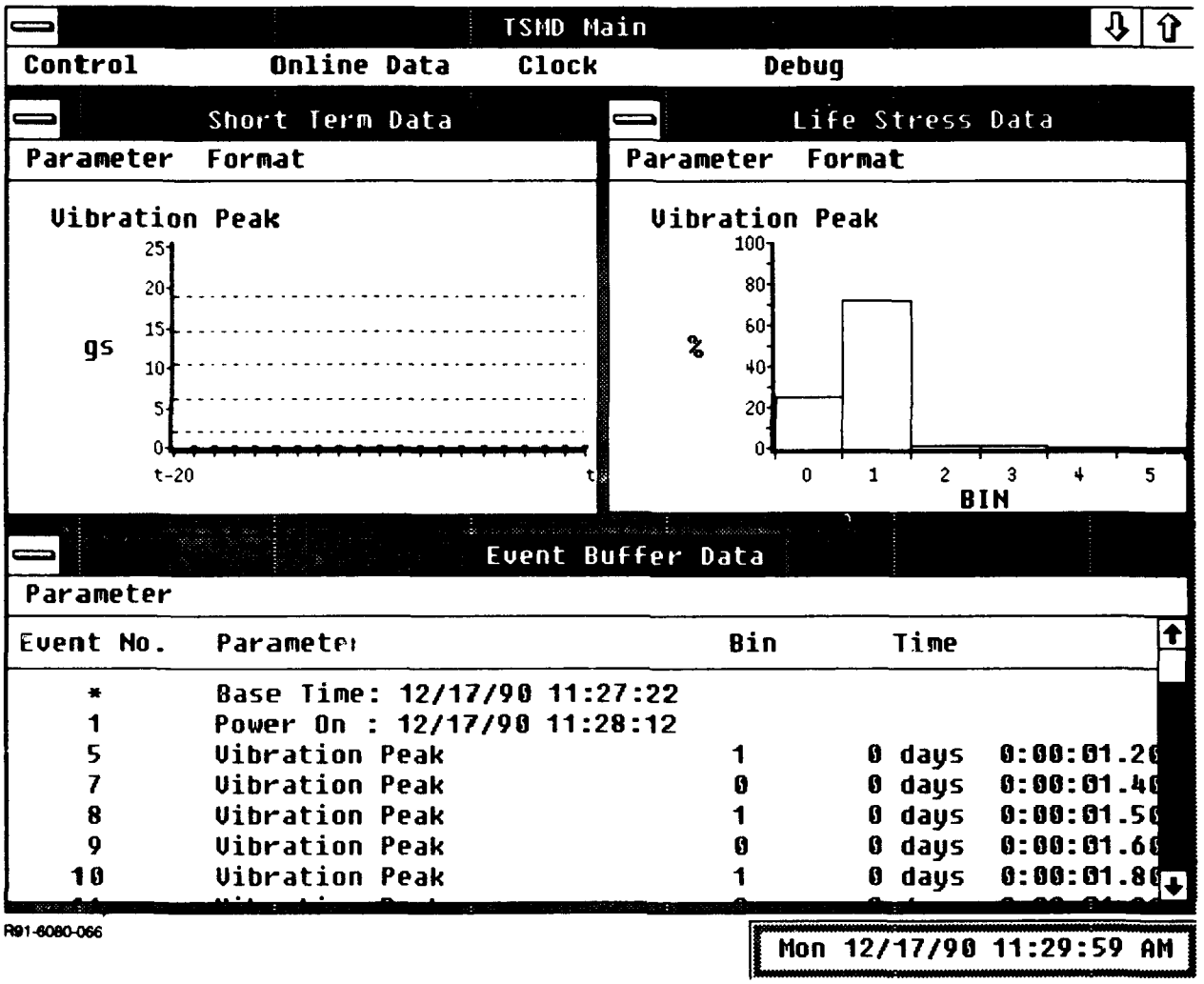



**Figure F-8. Scenario 2 TSMD window.**

B. Using the **[Pg Up]** key to scroli, observe that the Lisp Listener window contains the following:

0 Faulty Samples Misclassified.

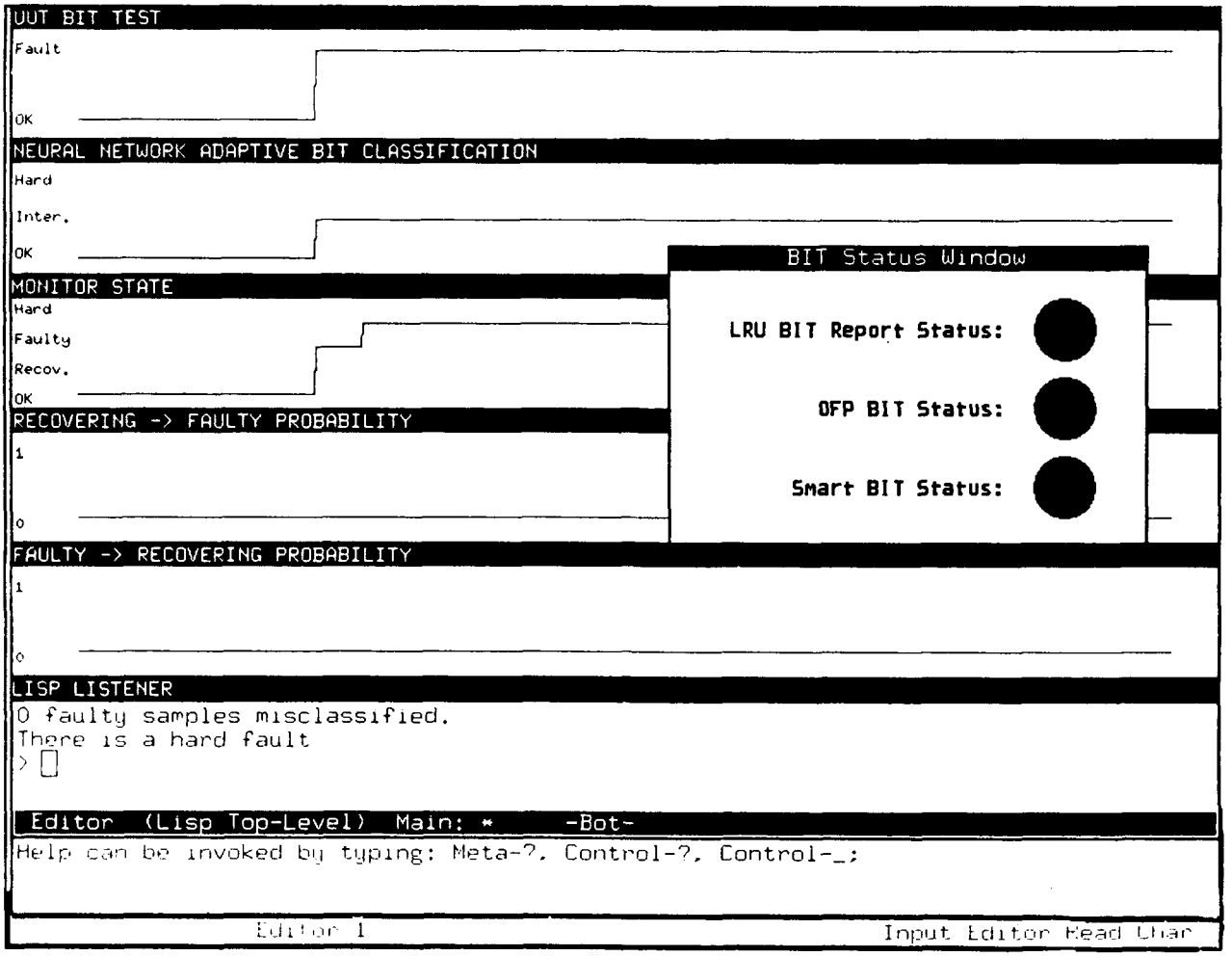
There is a hard fault

### F.2.4 Execute Scenario No. 3

1. Reposition the Smart BIT LISP LISTENER text cursor back to its proper place. At the Smart BIT computer keyboard, type the following:

**[Esc]**

**[Shift]-[>]**




R91-6080-067

**Figure F-9. Scenario 2 TSMD window.**

2. Re-display the last command on the Smart BIT LISP LISTENER window. At the Smart BIT computer keyboard, type the following:

   **[Esc]**

   **[p]**

3. Select a different Adaptive BIT paradigm on the Smart BIT computer LISP LISTENER. At the Smart BIT computer keyboard, type the following:

   A. Backspace over the text sequence:

      **neural-net)**

   B. Then type in:

      **knn)**

4. At the TSMD computer, display the TSMD System Control Dialog box by typing the following:

   **[Ctrl]-[s]**

5. Select to reset all previous TSMD data in memory:
   A. Move the TSMD computer mouse cursor (arrow) onto the TSMD computer System Control Dialog box Memory Resets group area and then over the box surrounding the word:

      ALL

   B. Click the TSMD computer left mouse button ONCE.
6. Extinguish the System Control Dialog box:
   A. Move the TSMD computer mouse cursor (arrow) onto the the bottom left of the System Control Dialog box over the box surrounding the word:

      OK

   B. Click the TSMD computer left mouse button ONCE.
7. Setup the UUT computer to run a scenario:
   A. Move the UUT computer mouse cursor (arrow) onto the menu bar (near the top of the UUT Main window) over the word:

      Run

   B. Click the UUT computer left mouse button ONCE.
8. Wait for the Run Dialog box to appear. Now select scenario no. 3 and run it:
   A. Move the UUT computer mouse cursor (in the file directory list of the Run Dialog box) over the scenario no. 1 filename:

      SCEN3.RUN

   B. Click the UUT computer left mouse button TWICE.
9. Wait for small UUT Window to appear at the center of the UUT computer screen (it will say: Running Scenario SCEN3.RUN). Scenario no. 3 is now running.

OPTION NOTE: Once running, if it is desired to stop the scenario before its normal end (abort), move the UUT computer mouse onto the small UUT window over the box surrounding the word: Cancel. Then click the UUT computer left mouse button ONCE.

10. Observe the following results at the TSMD computer:
    A. Observe that when the scenario begins, the Life Stress Data window display begins changing. The elapsed time displayed in the lower right corner of the screen will begin counting up, and a power-up event will be logged in the Event Buffer Data window.

B. Observe that when the Vibration Peak trace displayed in the Short-Term Data window crosses one of the thresholds drawn in red, an event is logged in the Event Buffer Data window.

C. Observe that the bar graphs displayed in the Life Stress Data window change in accordance with the data displayed in the Short-Term Data window. For example, when the Vibration Peak trace is in range 3, the bar for that range grows while the others shrink.

D. When the scenario ends, note that activity in the Life Stress Data and Event Buffer Data windows stops. The display should resemble that shown in Figure F-10.

11. Observe the following results at the Smart BIT computer:

A. At the end of the data analysis, observe that the window displayed agrees with the typical window shown in Figure F-11.

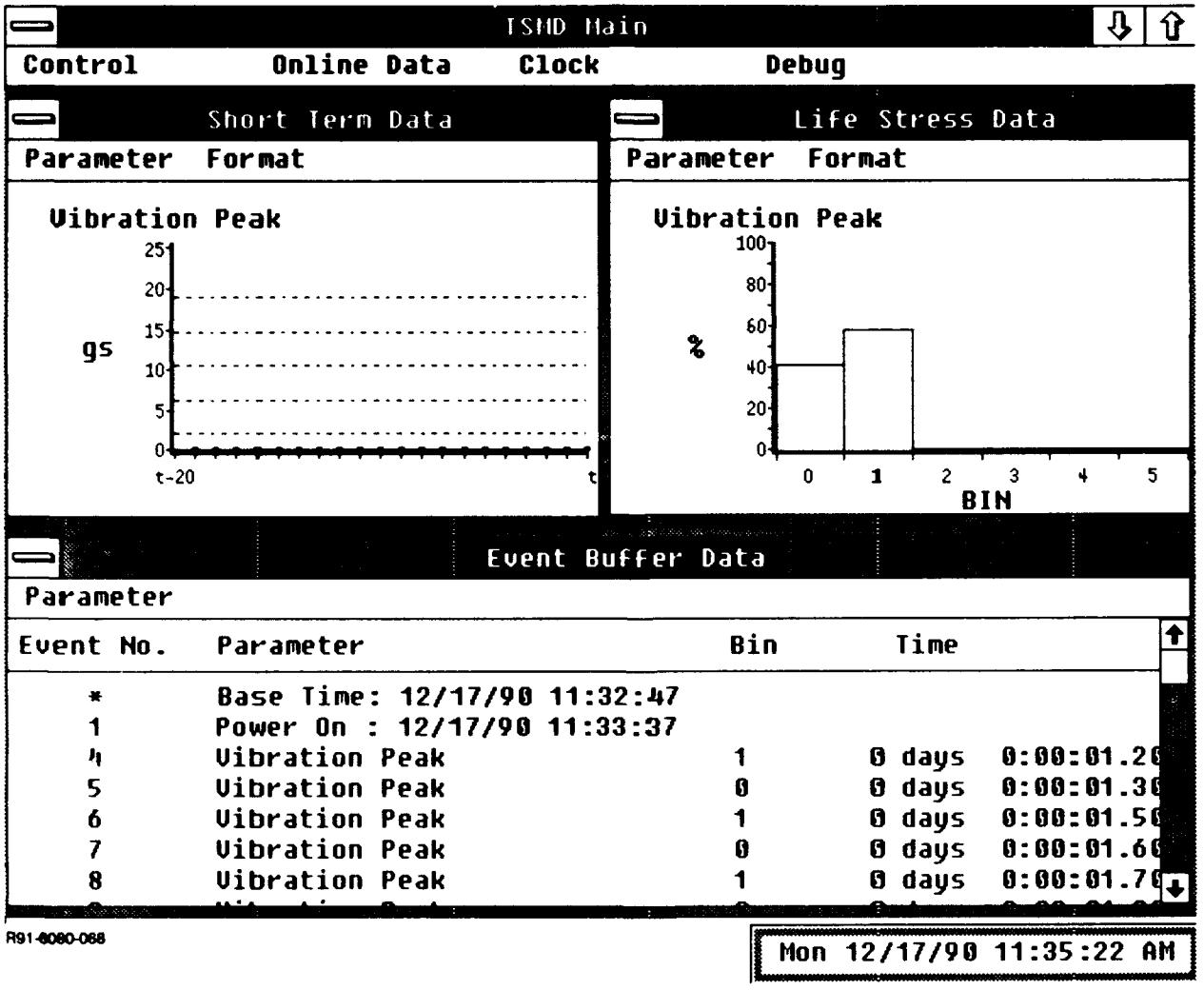
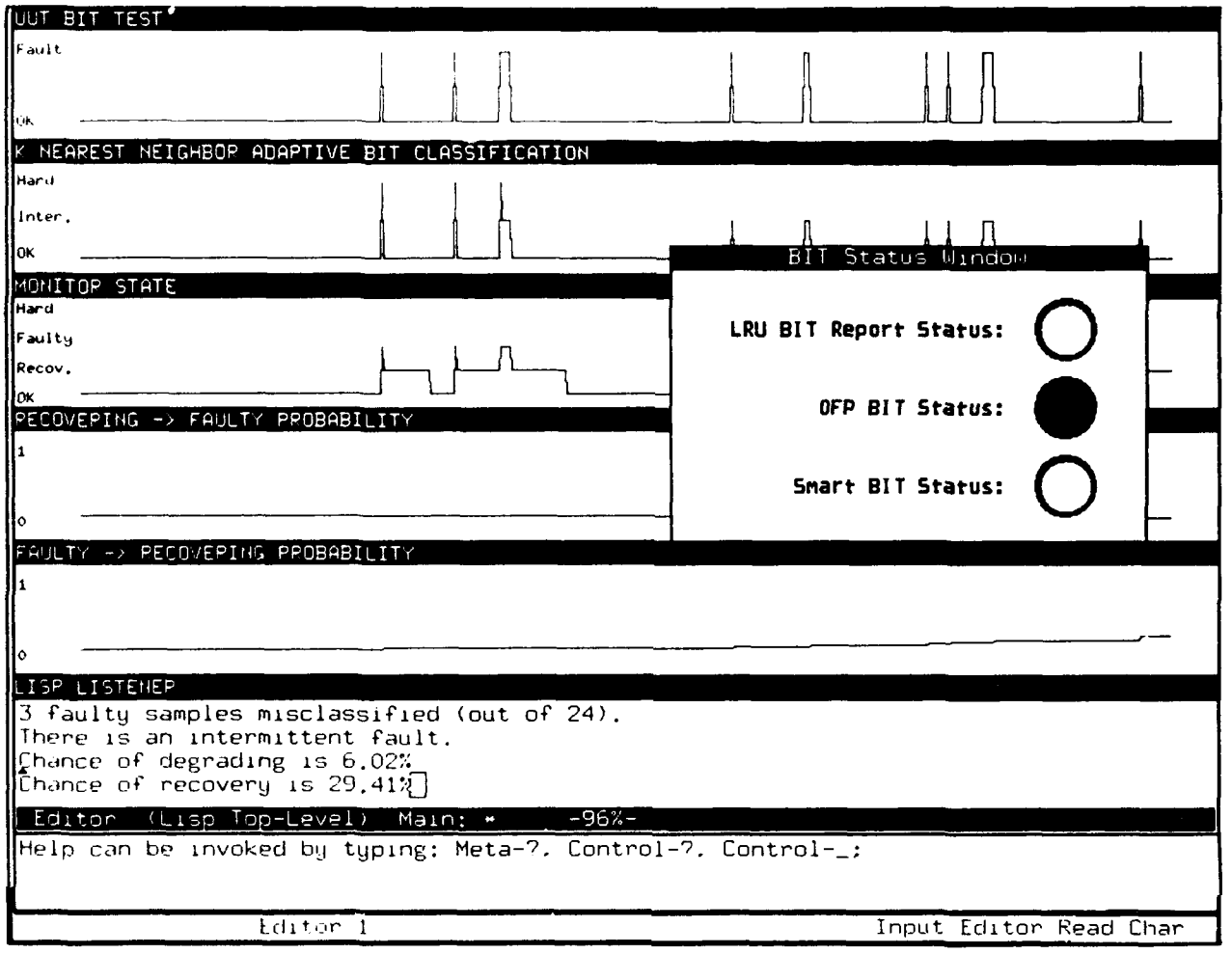



**Figure F-10. Scenario 3 TSMD window.**




1-6080-069

**Figure F-11. Scenario 3 Smart BIT window.**

NOTE: Because of the asynchronous nature of the Smart BIT / TSMD Integration System test bed, there may be differences between Figure F-11 and the actual window results.

B. Using the **[Pg Up]** to scroll, observe that the Lisp Listener window contains the following:

3 Faulty Samples Misclassified(out of 24)

There is an intermittent fault

Chance of degrading is 6.02 %

Chance of recovery is 29.4%

NOTE: Because of the asynchronous nature of the Smart BIT/TSMD Integration System testbed, there may be differences in percentages displayed versus those shown above. There may also be some differences in the display data.

## F.3 POWER SHUTDOWN FOR THE SMART BIT/TSMD INTEGRATION SYSTEM TEST BED

### F.3.1 Preparation Sequence for Shut-off of the UUT Computer

1. Leave the UUT computer applications program:
   A. At the UUT computer keyboard, type the following:
      **[Alt]-[F4]**
2. Leave the UUT computer Microsoft Windows program:
   A. At the UUT computer keyboard, type the following:
      **[Alt]-[F4]**
   B. At the UUT computer keyboard, type the following:
      **[Enter]**

### F.3.2 Preparation Sequence for Shut-off of the TSMD Computer

1. Leave the TSMD computer applications program:
   A. Move the TSMD computer mouse cursor over the TSMD Main title area and click the TSMD computer left mouse button.
   B. At the TSMD computer keyboard, type the following:
      **[Alt]-[F4]**
   C. At the TSMD computer keyboard, type the following:
      **[Alt]-[n]**
2. Leave the TSMD computer Microsoft Windows program:
   A. At the TSMD computer keyboard, type the following:
      **[Alt]-[F4]**
   B. At the TSMD computer keyboard, type the following:
      **[Enter]**

### F.3.3 Preparation Sequence for Shut-off of the Smart BIT Computer

1. Leave The Smart BIT applications program and X11 Windows environment:
   A. At the Smart BIT computer keyboard, type the following:
      **[Esc][Shift]-[>]**
   B. At the Smart BIT computer keyboard, type the following:
      **(exit)**

NOTE: After the X11 Window environment is aborted, there will be a plethora of warning messages; this is normal. Therefore IGNORE ALL WARNING MESSAGES.

2. Return to the top root Unix System V directory:
   At the Smart BIT computer keyboard, type the following:
   **cd /[ENTER]**
3. Start the Unix shutdown sequence:
   At the Smart BIT computer keyboard, type the following:
   **shutdown -y -g0[ENTER]**

NOTE: WAIT for the message on the Smart BIT Computer display to show:
Press any key to reboot

### F.3.4 Depress Power ON/OFF Switches to the OFF Position on all Computers and Monitors

## F.4 DEMONSTRATION OF UUT COMPUTER TIME LINE EDITORS

NOTE 1: Sections F.4 and F.5 are not required to demonstrate the scenario scripts. They are optional, and are provided as an additional guide to using the UUT computer editors.

NOTE 2: If a mistake is made while placing data into any of the Time Line Editors (temperature, accelerometer, or BIT), perform the following (while in the Time Line Editor) to erase the edited area:

1. Move the UUT computer mouse cursor onto the Time Line Editor menu bar and over the word:
   Edit
2. Click the UUT computer left mouse button ONCE.
3. Move the UUT computer mouse cursor onto the Edit menu and over the word:
   Clear Del
4. Click the UUT computer left mouse button ONCE.

### F.4.1 Demonstrate the UUT Computer Temperature Time Line Editor Tool

1. Select the Time Line Editor from the Tool menu:
   A. Move the UUT computer mouse cursor onto the UUT Main window menu bar and over the word:
      Tool

B. Click the UUT computer left mouse button ONCE.

C. Move the UUT computer mouse cursor onto the Tool menu over the words:

Time Line Editor

D. Click the UUT computer left mouse button ONCE.

NOTE: The rectangular temperature region (to the right of the y axis) is highlighted with a shade of amber to indicate that the Temperature Time Line Editor is active.

2. Begin drawing a sample temperature time line profile:

A. Move the UUT computer mouse cursor (arrow point) even with, and just to the right of, the 80-degree temperature mark on the y axis in the temperature region.

B. Press and hold down the UUT computer left mouse button, and drag the UUT computer mouse cursor to the -40-degree temperature mark at approximately the 20-second time mark. A line should be continually redraw as the UUT computer mouse cursor moves.

C. Release the UUT computer left mouse button and observe a fixed line is drawn from the point (0 second, 80 degrees) to the point (20 seconds,-40 degrees).

3. Continue to draw the temperature profile:

A. Move the UUT computer mouse cursor to the right of the end of the first temperature line drawn.

B. Press and hold down the UUT computer left mouse button and drag the UUT computer mouse cursor to the location (60 seconds,100 degrees) and release the UUT computer left mouse button.

C. Observe that a fixed line is drawn from the (20 seconds,-40 degrees) point to the (60 seconds,100 degrees) point.

### F.4.2 Demonstrate the UUT Computer Accelerometer Time Line Editor Tool

1. Select the accelerometer time line editor:

A. Move the UUT computer mouse cursor to the Time Line Editor menu bar and over the word:

Sensor

B. Click the UUT computer left mouse button ONCE.

C. Move the UUT computer mouse cursor onto the Sensor menu over the last icon down in the left column:

D. Click the UUT computer left mouse button ONCE.

NOTE: The rectangular accelerometer region (to the right of the y axis and the middle area of the display) is highlighted with a shade of amber to indicate that the Accelerometer Time Line Editor is now the active region to edit.

2. Place the damped sinusoid accelerometer event (one second in duration) at the 10-second time point (perform the following):
   A. Move the UUT computer mouse cursor into the highlighted accelerometer region.
   B. Press and hold down the UUT computer left mouse button, and line up the vertical cursor line over the 10-second mark.
   C. Release the UUT computer mouse button and observe that the damped sinusoid icon appears at the 10-second time location.

### F.4.3 Demonstrate the UUT Computer BIT Time Line Editor Tool

1. Select the BIT time line editor:
   A. Move the UUT computer mouse cursor to the Time Line Editor menu bar and over the word:
      BIT
   B. Click the UUT computer left mouse button ONCE.
   C. Move the UUT computer mouse cursor onto the BIT menu over the word:
      OK
   D. Click the UUT computer left mouse button ONCE.

NOTE: The rectangular BIT Status region (to the right of the y axis and the bottom area of the display) is highlighted with a shade of amber to indicate that the BIT Time Line Editor is now the active region to edit.

2. Draw the BIT Status OK state for 10 seconds:
   A. Move the UUT computer mouse cursor so that it is over the 10-second mark in the BIT Status highlighted region.
   B. Click the UUT computer left mouse button and observe that a horizontal line appears, indicating the OK state from 0 to 10 seconds.
3. Select the Fault BIT Status:
   A. Move the UUT computer mouse cursor to the Time Line Editor menu bar and over the word:
      BIT
   B. Click the UUT computer left mouse button ONCE.

C. Move the UUT computer mouse cursor onto the BIT menu over the word:

Fault

D. Click the UUT computer left mouse button ONCE.

4. Draw the BIT Status Fault state at the 10 to 15 seconds:
   A. Move the UUT computer mouse cursor so that it is over the 10-second mark in the BIT Status highlighted region.
   B. Click the UUT computer left mouse button and drag the UUT computer mouse to the right to the 15-second mark.
   C. Release the UUT computer left mouse button.
   D. Observe that a pulse has been drawn in for a BIT Fault lasting 5 seconds starting at the 10-second mark and ending at the 15-second mark.

NOTE: The pulse that was drawn represents 5 seconds of consecutive LRU BIT fault reports starting at the scenario time of 10 seconds and ending at the 15-second scenario time mark.

5. Draw an impulse BIT fault report (one BIT fault report):
   A. Move the UUT computer mouse cursor so that it is over the 20-second mark in the BIT Status highlighted region.
   B. Then Click the UUT computer left mouse button ONCE.

NOTE: Observe that a single BIT fault report is drawn at the 20-second mark.

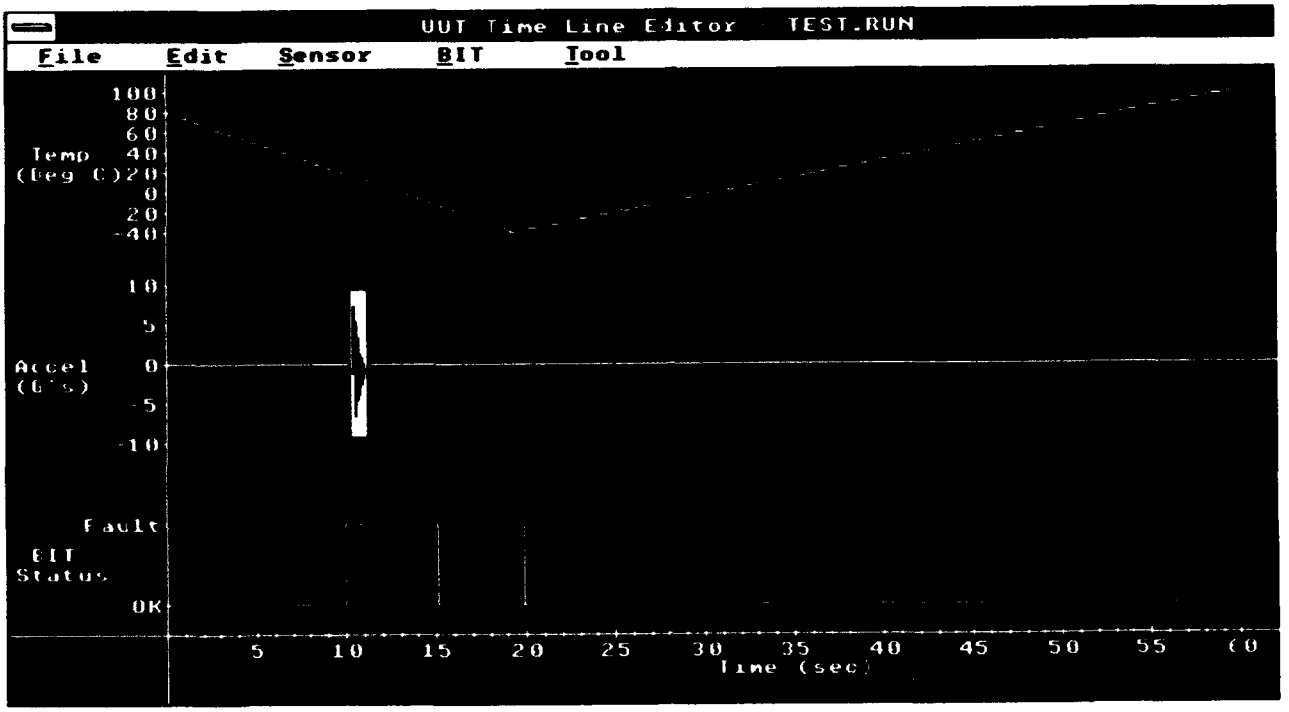
6. Select the OK BIT Status:
   A. Move the UUT computer mouse cursor to the Time Line Editor menu bar and over the word:

      BIT

   B. Click the UUT computer left mouse button ONCE.
   C. Move the UUT computer mouse cursor onto the BIT menu over the word:

      OK

   D. Click the UUT computer left mouse button ONCE.
7. Draw the BIT Status OK state from the 20-second to 60-second mark:
   A. Move the UUT computer mouse cursor so that it is over the 60-second mark in the BIT Status highlighted region.
   B. Click the UUT computer left mouse button and observe that a horizontal line appears, indicating the OK state from 20 to 60 seconds.
8. Observe that the Time Line Editor display appears like Figure F-12.




R91-6080-070

**Figure F-12. UUT Time line display.**

### F.4.4 Save the Data in the Three Time Line Editors to a File, and Then Restore Those Data to the Three Time Line Editors from the File.

1. Display the Save As menu selection:
   A. Move the UUT computer mouse cursor onto the Time Line Editor menu bar over the word:
      File
   B. Click the UUT computer left mouse button ONCE.
   C. Move the UUT computer mouse cursor onto the File menu over the word:
      Save As...
   D. Click the UUT computer left mouse button ONCE.

   NOTE: A Save File dialog box appears.

2. Save the Time Line Editor data:
   A. At the UUT computer keyboard, type in:
      **test.run**
   B. Move the UUT computer mouse cursor over the box surrounding the word:
      Save

C. Click the UUT computer left mouse button ONCE.

NOTE: Time Line Editor information is now saved in test.run file data.

3. Clear the Time Line Editor Display:

A. Move the UUT computer mouse cursor onto the Time Line Editor menu bar over the word:

File

B. Click the UUT computer left mouse button ONCE.

C. Move the UUT computer mouse cursor onto the File menu over the word:

New

D. Click the UUT computer left mouse button ONCE.

NOTE: All temperature, accelerometer, and BIT data in the Time Line Editors are erased.

4. Restore the Time Line Editor data from the test.run file:

A. Move the UUT computer mouse cursor onto the Time Line Editor menu bar over the word:

File

B. Click the UUT computer left mouse button ONCE.

C. Move the UUT computer mouse cursor onto the File menu over the word:

Open

D. Click the UUT computer left mouse button ONCE.

NOTE 1: Observe that an Open File dialog box is displayed.

E. Move the UUT computer mouse cursor onto the Open File dialog box file list and over the file name:

test.run

F. Click the UUT computer left mouse button TWICE.

NOTE 2: Observe that the temperature, accelerometer, and BIT Status data are redisplayed in their respective Time Line Editors.

## F.5 DEMONSTRATION OF UUT COMPUTER SCENARIO EDITOR

NOTE: In order to run the test.run file scenario, Subsections F.4 and F.5 MUST be performed.

### F.5.1 Select the Scenario Editor Tool

1. Move the UUT computer mouse cursor onto the menu bar over the word:

Tool

2. Click the UUT computer left mouse button ONCE.
3. Move the UUT computer mouse cursor onto the Tool menu over the word:

   Scenario Editor
4. Click the UUT computer left mouse button ONCE.

NOTE: Window title now shows the Scenario Editor.

### F.5.2 Open the Scenario No. 1 Data File into the Scenario Editor

1. Move the UUT computer mouse cursor onto the Scenario Editor menu bar over the word:

   File
2. Click the UUT computer left mouse button ONCE.
3. Move the UUT computer mouse cursor onto the File menu over the word:

   Open
4. Click the UUT computer left mouse button ONCE.

NOTE: Observe that an Open File dialog box is displayed.

5. Move the UUT computer mouse cursor onto the Open File dialog box file list and over the file name:

   scen1.dat
6. Click the UUT computer left mouse button TWICE.

NOTE 1: Observe that the SCADC parametric and BIT data are displayed in the Scenario Editor.

NOTE 2: Once a Scenario Data file is opened, the Scenario Editor can be used to delete (use backspace key) or add text like any text editor.

### F.5.3 Save the SCEN1.DAT Data as TEST.DAT File Name into the TEST.RUN FILE

1. Display the Save As menu selection:
   A. Move the UUT computer mouse cursor onto the Time Line Editor menu bar over the word:

      File
   B. Click the UUT computer left mouse button ONCE.
   C. Move the UUT computer mouse cursor onto the File menu over the word:

      Save As...
   D. Click the UUT computer left mouse button ONCE.

NOTE: A Save File dialog box appears.

### F.5.4 Save the Scenario Editor Data

1. At the UUT computer keyboard, type in:

   test.dat

2. Move the UUT computer mouse cursor on the Save As dialog box over the box surrounding the word:

   Save

3. Click the UUT computer left mouse button ONCE.

NOTE: A Save Data File Name Into RUN File dialog box appears.

4. Move the UUT computer mouse cursor on the Save Data File Name Into RUN File dialog box over the box surrounding the word:

   Save

5. Click the UUT computer left mouse button ONCE.

NOTE: Scenario Generation Editor file information is now saved in test.run file.

### F.5.5 Return to the UUT Main Window

1. Move the UUT computer mouse cursor onto the menu bar over the word:

   Tool

2. Click the UUT computer left mouse button ONCE.
3. Move the UUT computer mouse cursor onto the File menu over the word:

   UUT Main

4. Click the UUT computer left mouse button ONCE.

NOTE: Window title now shows UUT Main.

# APPENDIX G

# VIBRATION SOURCES AND RELATIONSHIPS
# FINAL REPORT EXCERPT; MARCH 1989
# AFSC CONTRACT NO. F30602-87-D-0152

## G.1 VIBRATION SOURCES AND RELATIONSHIPS

The development and effective deployment of a TSMD requires an understanding of the mechanisms of vibration response due to interaction of the host vehicle with typical environments, the probability of occurrence of each type of stress, and the influences of those stresses on the ultimate life of the device itself. The discussions herein will concentrate on TSMD applications for jet aircraft on typical missions. Similar arguments can, however, be developed for other types of vehicles and specific mission types.

Vibration responses within jet aircraft or within captive external stores are caused by a number of phenomena. The principal sources of steady state response are the passage of turbulent airflow over external surfaces and the acoustic noise emanating from the complex pressure fields generated by the jet and fan engine exhaust. When the pressure fields from these sources impinge on external surfaces during flight, random vibration responses developed at the surface propagate throughout the structure and internal systems. Thus, the internal equipment vibration response levels and spectrum shapes can be directly related to the interaction of the aircraft with the environment as modified by the transmissibility of the intermediate structure. The principal sources of transient vibration include takeoff and landing transients, gunfire, and short-term repositioning of turbulent airflow as caused during speed brake actions and maneuvers.

For jet aircraft flying at semi-steady state conditions of altitude and velocity, the overall vibration response level at a particular physical location is roughly proportional to dynamic pressure, with a distinct discontinuity in the transonic region (Mach number from 0.9 to 1.0, approximately). Dynamic pressure is directly proportional to the air density and pressure at altitude and to the square of the aircraft velocity. For any specific aircraft velocity, the dynamic

pressure decreases with increased altitude. For a fixed altitude, dynamic pressure increases rapidly with aircraft velocity. Since the velocity of sound decreases with increasing altitude, the aircraft velocity at which Mach = 1 is reached also decreases with increased altitude.

The dynamic pressure and the resultant vibration response levels depend on the operational envelope of a particular aircraft, which in turn depends to a large extent on the drag and turbulence caused by external stores and their locations on the aircraft.

Selection of a TSMD vibration trigger threshold for a particular application must be tailored to the particular equipment configuration, to the mission type, to the location of the sensor within the aircraft or captive external store, and to the particular purpose for data collection flights. For example, the constant of proportionality of RMS vibration level to dynamic pressure may vary by a factor of 10 depending on the physical orientation and location of a accelerometer sensor. The overall vibration level on electronic equipment may range from 0.7 to 7 g rms at a dynamic pressure of 1200 $lb/ft^2$, with the higher level being exceeded much less often than 5% of the time, even during transients such as speed brake actions.

The spectrum shape of vibration response as a function of frequency is relatively constant except for overall spectral level. Figure G-1 shows spectra which were acquired at four different dynamic pressures which are relatively constant in shape at frequencies above 100 Hz. The large low-frequency maxima at 48 Hz on the curves indicate varying degrees of buffet response, with the solid curve showing a worst-case, low-altitude, rigid-body motion in the transonic region.

Figure G-2 shows typical response spectrum variations which result from local structural response and transmission path effects at sensor locations ranging from the aircraft structure to a circuit board within a subassembly within a representative electronic assembly. The vibration spectrum and overall vibration level at the sensor location on the printed wiring assembly obviously requires a different constant of proportionality to relate vibration level to dynamic pressure than for a sensor attached to the aircraft structure. If the transfer functions between the two locations was known, however, then information at one location could be inferred from data at the other.

Aircraft and the equipment they carry are designed to provide a useful life greater than some contractual minimum number of hours or missions. All possible types of mission, their approxi-

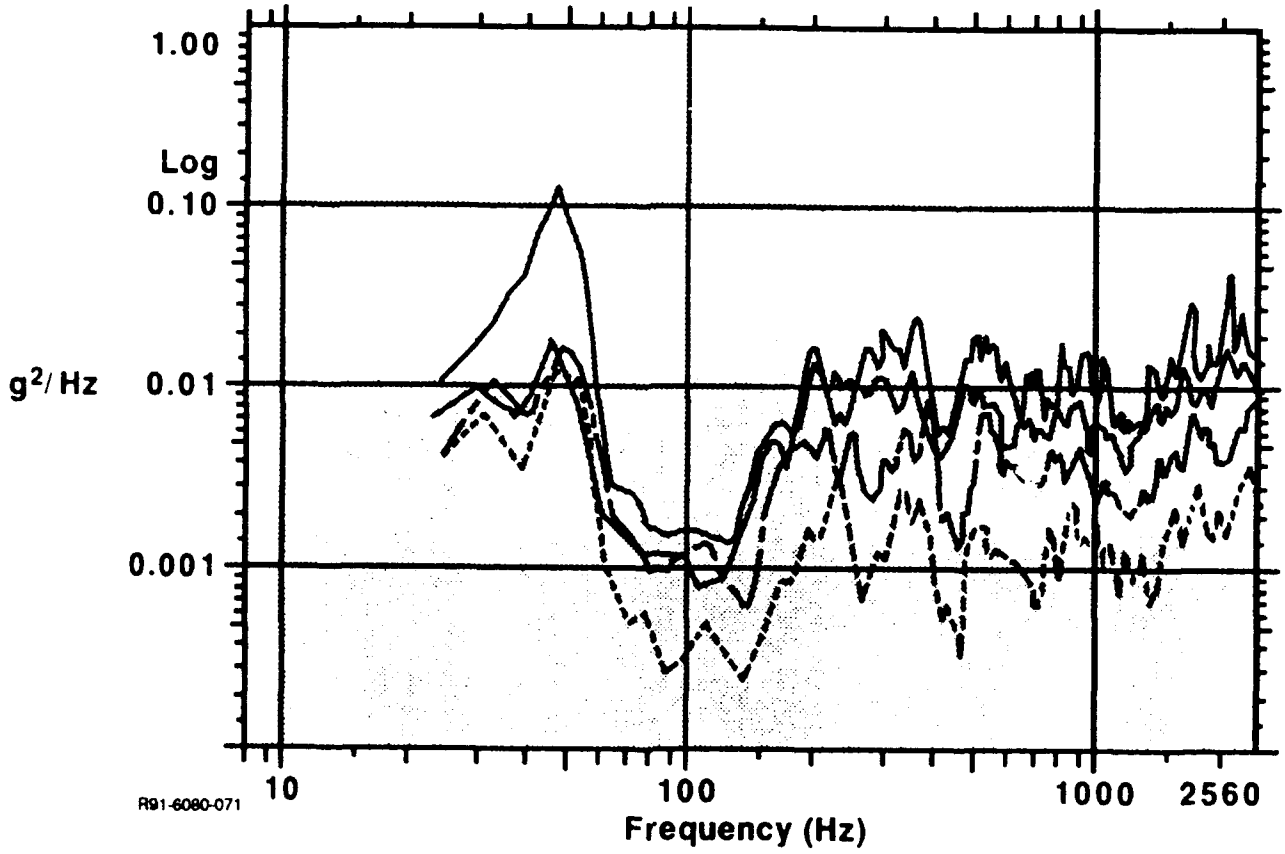



Figure G-1. Measured vibration spectra at form levels of dynamic pressure.

mate duration, and the fraction of time spent in each portion of the corresponding mission profile (takeoff, cruise, low-level attack, etc) are defined. From the relative mix of missions and the details of each mission type, the resulting stresses and the cumulative contribution to fatigue damage throughout the life of the aircraft can be estimated. Design of an aircraft and of the associated preventive maintenance and inspection program can then be undertaken. During early field deployment, flight records are compiled to develop actual flight time as a function of altitude and Mach number, as shown in Figure G-3.

Each block of Figure G-3 corresponds to the number of hours of a total of 6,000 which were spent in a particular range of altitude and Mach number. Also shown in Figure G-3 are the maximum permissible envelope and a more representative maximum operational envelope. The sloped portions of the envelopes which terminate just above Mach 1 are roughly consistent with constant dynamic pressure lines at 1400 $lb/ft^2$ for the operational envelope and 1800 $lb/ft^2$ for the maximum permissible enveloped. A dynamic pressure of 1200 $lb/ft^2$ is a more realistic envelope

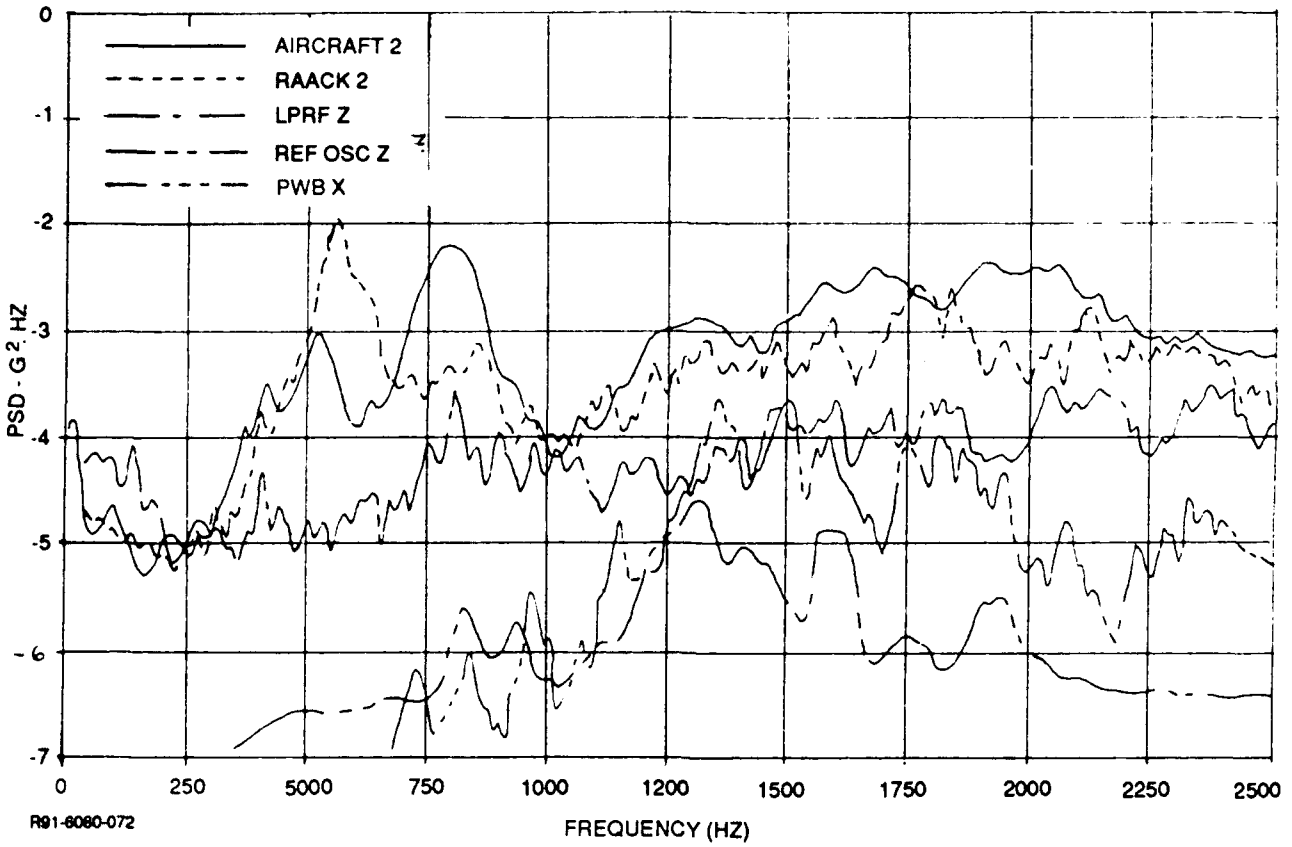



**Figure G-2. Acceleration spectral densities (G /Hz) at five locations during F-16 flight.**

limit for an aircraft with a load of external stores such as fuel tanks, instrument pods, or weapons. More than 60% of the flight hours shown in Figure G-3 are spent at dynamic pressures in the range of 300 to 350 lb/ft$^2$. Similarly, less than 3% are spent at dynamic pressure of 1200 lb/ft$^2$ or greater. The 136 hours in the block with limits of M = 0.8 to 1.0 and 0 to 10,000 ft altitude represents the worst-case steady environmental vibration with any appreciable frequency of occurrence, exceeded only for a very small percentage of time. Vibration-induced equipment failures are very infrequent at low levels of stress (300 to 350 lb/ft$^2$), with a high frequency of occurrence. Typically more than 50% of all vibration-induced equipment failures occur at and above a dynamic pressure of 1000 lb/ft$^2$. It is also worthwhile noting that high G loads which may stress the pilot or the airframe do not necessarily result in high random vibration levels. In addition, the peak levels of random vibration are on the order of three times the RMS overall levels and, under worst-case steady state conditions, frequently exceed the infrequent 9 g short duration static load which result from maneuvers.

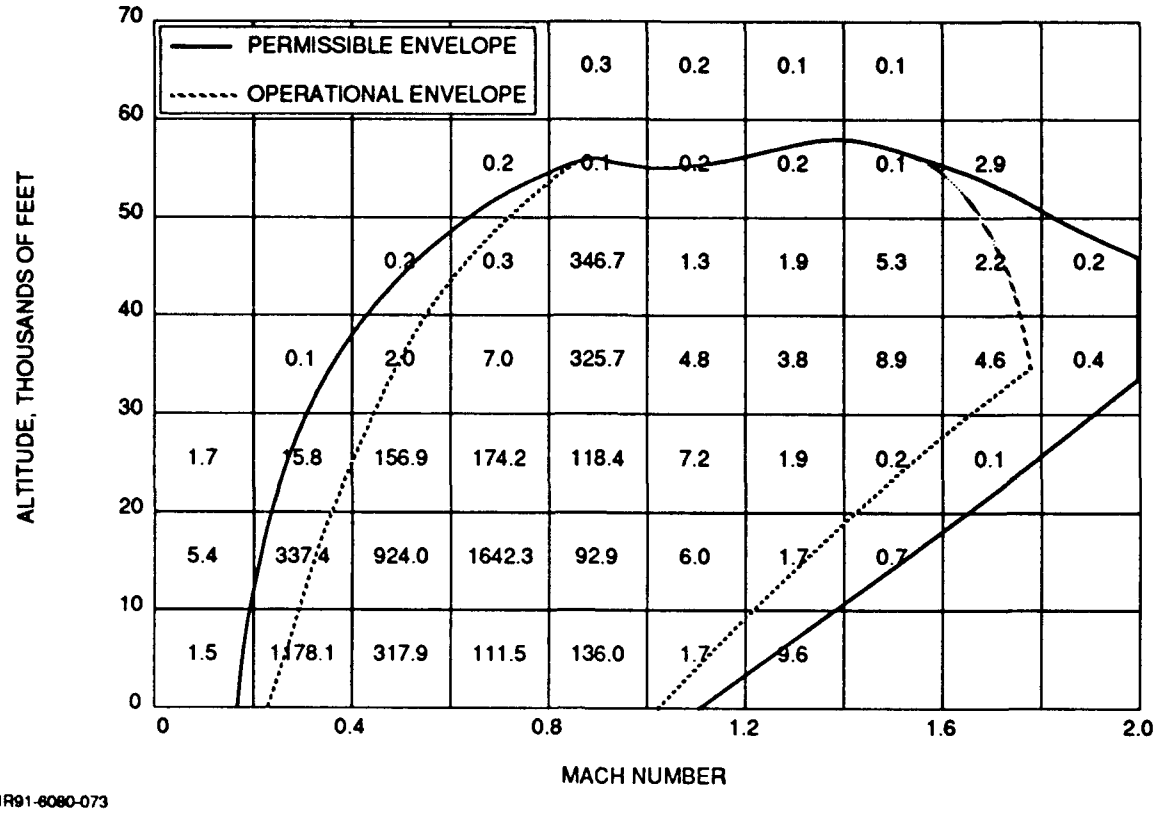
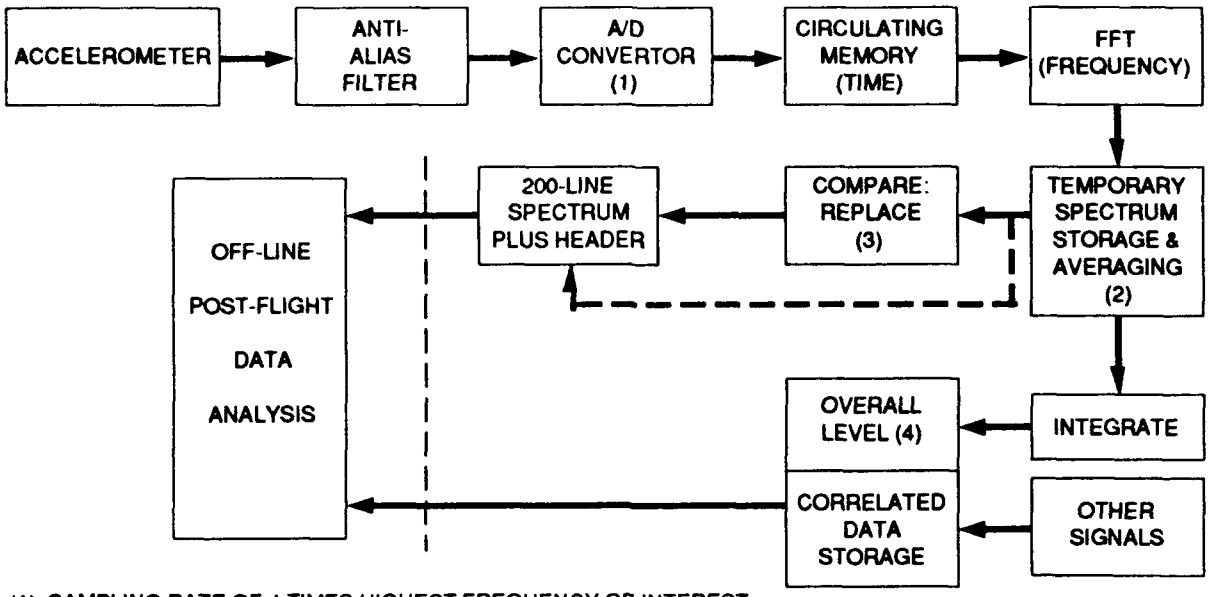



**Figure G-3.** **Number of hours within specific altitude-Mach number segments over 6,000 flight hours for a typical fighter aircraft.**

## G.2 VIBRATION DATA ACQUISITION CONSIDERATIONS

With a limited number of data records available in TSMD EPROM, an appreciable amount of data compaction of vibration signals is required. If wake up circuitry is triggered by a vibration peak which exceeds a present threshold, circuitry can be relatively simple. If wake up circuitry is triggered by a built-in test failure, the correlated peak vibration level (as modified by attack and decay time constants) provides an additional data point. For many such events a statistical picture could be developed, but it would take many flights or many hours to acquire many events.

A more sophisticated data acquisition and compaction system could be developed which would retain both spectrum shapes and overall level within limited data storage constraints. Figure G-4 is one concept of such a system. An A/D converter running at a sampling rate of four times the highest frequency of interest could continuously store time history data into a circulat-




(1) SAMPLING RATE OF 4 TIMES HIGHEST FREQUENCY OF INTEREST

(2) AVERAGE OF 16 INDEPENDENT TIME HISTORIES RESULTS IN ERROR ≤ ± 2 OF B

(3) TEMPORARILY STORED AVERAGE SPECTRUM IS COMPARED WITH 200–LINE SPECTRUM. IF LARGER AT ALL FREQUENCIES, REPLACE WITH NEW SPECTRUM & HEADER INFORMATION (ALTITUDE, SPEED, TIME, ETC)

(4) OVERALL LEVEL AS ASSOCIATED WITH FLIGHT CONDITIONS IS A SOURCE FOR MANY OFF-LINE ANALYSIS USES.

MR91-6080-074

***Figure G-4. Possible Vibration Data Acquisition and Compaction System.***

ing memory. The converter would be preceded by gain adjustment circuitry and an antialiasing filter to avoid digitization error. Data could be extracted from the circulating memory and applied to a digital fast Fourier transform and be converted to the frequency domain. Up to 15 successive spectra could be averaged into a temporary storage area. An average of 16 independent time histories results in a spectral error of approximately ± 2 dB. (The same approach could be applied to a helicopter where the vibration environment consists of a number of low-fre-quency sinusoids superimposed on a broadbound random base. For a helicopter, however, accurate definition of the sinusoids would require a spectral resolution on the order of 1 Hz or better.) The first temporarily stored average spectrum could be stored in a 200-line file plus header information associated with flight conditions, such as altitude and speed. The average spectrum could be integrated and stored as an overall RMS vibration level. If succeeding averaged spectra exceeded the stored 200-line spectrum, they would replace it. If succeeding spectra did not exceed the 200-line spectrum it would remain unchanged and could be recovered as the worst-case environment encountered. If environmental temperature and vibration levels could be

correlated with aircraft conditions such as altitude, velocity, and altitude in addition to BIT failures and time, many useful kinds of post-flight data analysis could be accomplished. Sets of correlated data could be used to develop mission profiles, to enumerate failure type and frequency as functions of flight conditions, and to develop flight environment simulations which could be used in the laboratory to verify fixes indicated by bit failures or to perform combined environment reliability testing. With sufficient knowledge of transfer functions between the TSMD sensors and other locations, stress levels at those locations could be estimated. By relatively minor variations to a sophisticated data acquisition system, vibrations resulting from gunfire and maneuver actions which, due to their transient nature, require a different analysis approach could also be acquired and stored.

## G. 3 TSMD RELIABILITY CONSIDERATIONS

In order for a TSMD to be an effective tool, the design and physical realization must be capable of a reliable long life in the worst-case temperature and vibration environments likely to be encountered in potential use locations. Installations within a captive external store, such as an ECM or instrumentation pod, is likely to result in worst-case extremes in both vibration level and temperature.

In addition to careful placement of a TSMD on a circuit board to avoid excessive temperature excursions, consideration must also be given to placement on a circuit board to obtain a long life in terms of vibration environment exposure. When subjected to vibration exposure, a circuit board bends at one or more natural bending frequencies which are dependent on the board size, stiffness, and type of support at the edges. Figure G-5 shows exaggerated mode shapes for boards supported at three or four sides. For a board supported on four sides, the center of the board is cyclically displaced symmetrically from the resting position at the lowest natural bending frequency in either the short or long dimensions. For a board supported on three sides, the maximum displacement point moves toward the free edge. A rigid component (TSMD) mounted at the center of a board has its attachment leads cyclically bent as the board bends at its natural frequencies. For a given center of board displacement, the relative displacement between the board and the edge of the component is a function of component size, with a larger component edge displacement resulting from a larger component. Figure G-6 shows how the relative displacement at the edge of a component varies with component size and position with respect to the circuit board center. The larger the component edge displacement, the larger will be the flexing of a component attachment lead and the larger will be the stress on that lead. The lead

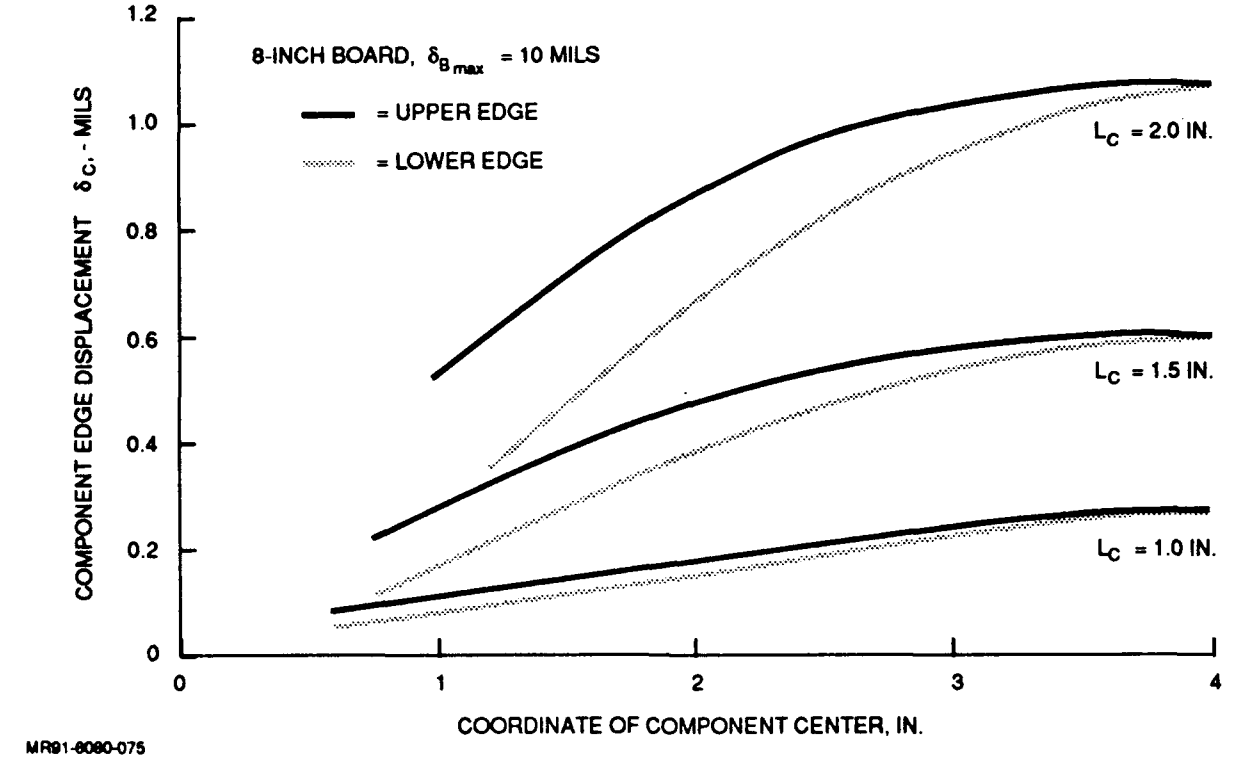



**Figure G-5. Exaggerated mode shapes of printed wiring boards.**

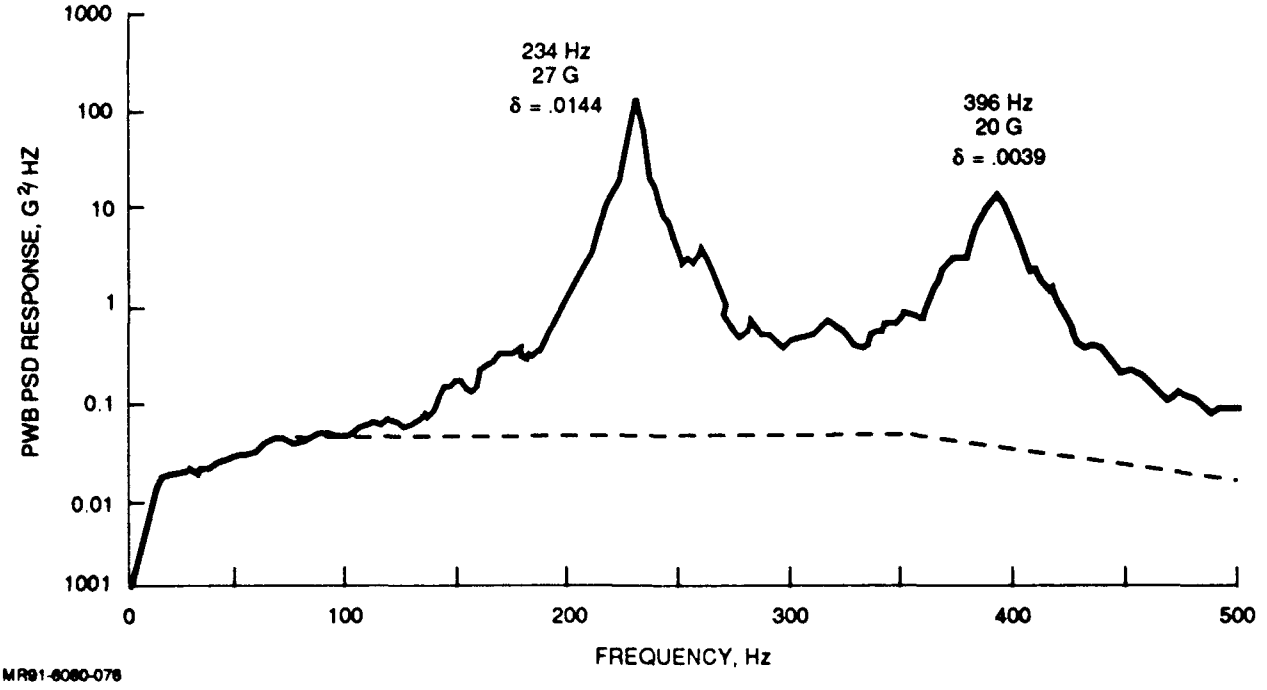



**Fig. G-6 Relative component edge displacement for various position of three component sizes mounted to an 8-in. circuit board with a center displacement of 0.01 in.**

stress is also a function of the lead dimensions and configuration. If the peak lead stress can be kept below the endurance strength of the attachment lead material, the lead will survive an infinite number of cyclic bends. If the peak lead stress exceeds the endurance strength, the lead will fail by fatigue at some finite time.

For a given board size and TSMD position, the center of board displacement depends on both the excitation level and the board transmissiblity. Figure G-7 shows an example of the spectral response of a circuit board. Magnification of response with respect to input at two bending frequencies is obvious. One or more stiffeners at the center and free edge could increase the natural frequencies and significantly reduce the magnification, resulting in a much smaller displacement and a much longer component lead life.

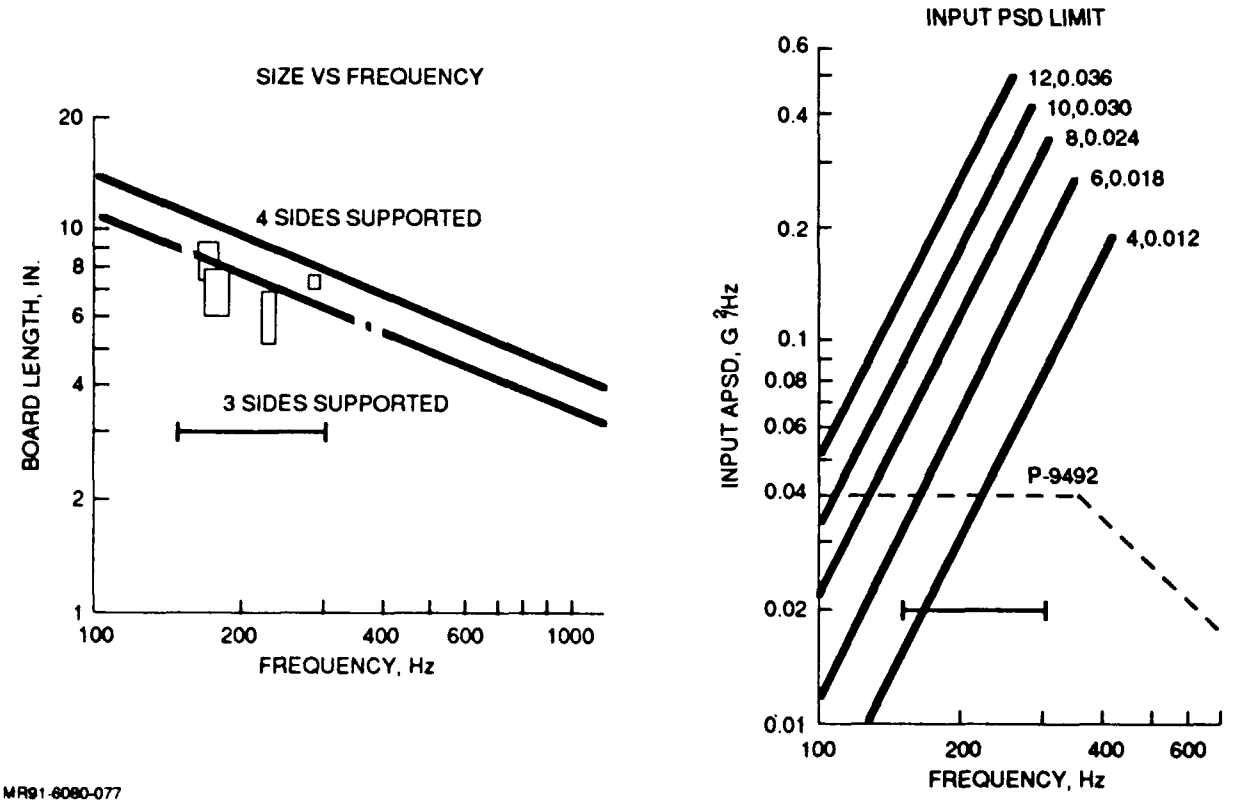



**Fig. G-7. Examples of vibration response spectrum for an 8-in. circuit board supported on three sides with an accelerometer located near the center of the free edge.**

Figure G-7 also demonstrates the importance of physical location of a TSMD acceleration sensor. Placement near the maximum displacement region of a circuit board results in a spectral sensitivity to the lowest natural bending frequency. Data at that location are dominated by board

response, difficult to relate to the excitation spectrum, and difficult to predict. The magnification of a circuit board is very strong function of the method of support at the edges, and may vary over a wide range for some types of board guide or edge damping method. A far better physical location for a TSMD acceleration sensor would be near a corner with both adjacent edges supported. Response spectra at such a location are typically much closer to the excitation spectra than for any other location.

Figure G-8 shows examples of theoretical and measured natural frequencies for square circuit boards supported on three or four sides. Figure G-8 also shows the maximum safe board displacement for a range of board sizes. The plots represent the maximum input power spectral density which can be applied versus first bending frequency for each size board. One-inch-square components with conventional attachment leads were used to perform the example calculations.

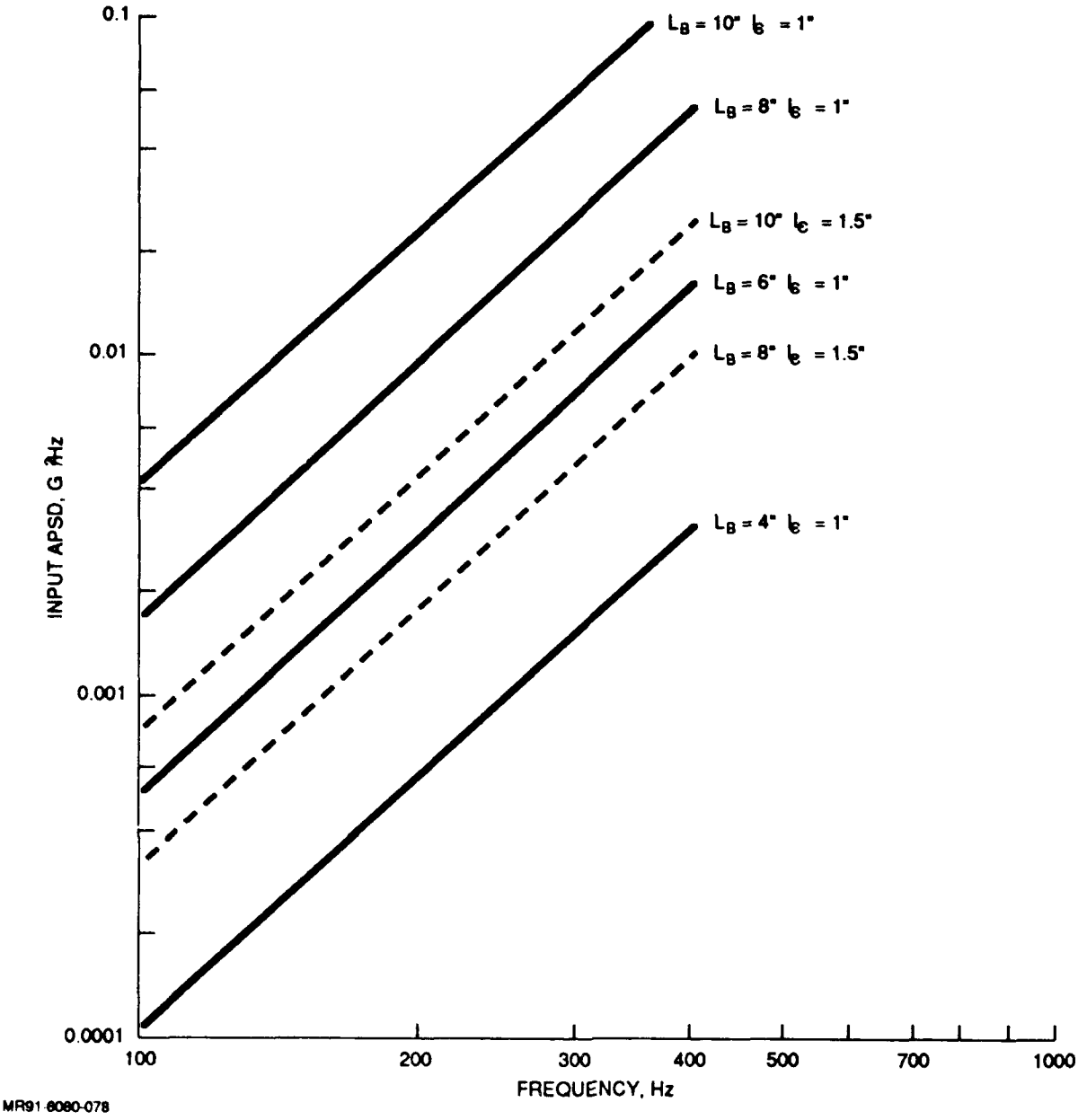



**Fig. G-8. Examples of natural frequency as a function of circuit board size. Maximum safe input PSD and displacement for typical board sizes.**

## *MISSION*

## *OF*

## *ROME LABORATORY*

*Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence ($C^3I$) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of $C^3I$ systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.*